HyperCLOVA X THINK

NAVER Cloud
HyperCLOVA X Team

Abstract

We introduce HyperCLOVA X THINK, the first reasoning-focused large language model in the HyperCLOVA X family, pre-trained on roughly 6 trillion high-quality Korean, and English tokens, augmented with targeted synthetic Korean data. It was implemented as a compute-memory-balanced Peri-LN Transformer scaled with μP, pre-trained through a three-stage curriculum that expands the context window to 128K tokens, and post-trained via supervised fine-tuning with Reinforcement Learning from Verifiable Rewards supports both detailed rationale and concise-answer modes. It delivers competitive performance against similarly sized models on Korea-focused benchmarks such as KMMLU, CSAT, KoBALT-700, HAERAE-1.0, and KoBigBench, while preserving robust bilingual consistency and translation quality. In addition, a vision-augmented variant matches or exceeds GPT-4.1 on the KCSAT STEM benchmark, all of which are achieved with substantially lower training compute than existing models of similar sizes. We also present a pruning and distillation technique that will soon be applied to HyperCLOVA X THINK for an open-source and business-friendly foundation model. Altogether, these capabilities position HyperCLOVA X THINK as a robust foundation for Korean AI innovation and a valuable resource for the global research community.

1 Introduction

Recent advancements of large language models (LLMs) have drawn increased attention to their reasoning abilities, going beyond simple memorization of factual knowledge to deriving logical conclusions. Models like GPT-o1 (OpenAI et al., 2024b), R1 (DeepSeek-AI et al., 2025), and QwQ (Qwen Team, 2025) exemplify such effort, demonstrating that the ability to perform logical inferences and multi-step problem solving can significantly broaden the scope of AI applications.

At the same time, the notion of sovereign AI is being established as an important goal. As LLMs continue to be deployed in various regions around the globe, the need for linguistic fluency and cultural sensitivity tailored toward a given region, as well as data governance that aligns with regional values and regulations, is growing. In this regard, our immediate focus is Korea.

To meet the imperatives of both advanced reasoning and sovereign AI—for Korea, in particular—we present HyperCLOVA X THINK (*henceforth* THINK). It was trained via a strategic preparation of training data and use of the latest pre- and post-training techniques.

In particular, we curated a corpus of roughly six trillion tokens that balances high-quality Korean and English text with targeted synthetic Korean data. This mixture improves linguistic breadth while safeguarding cultural and domain relevance. The model architecture follows a compute-memory-balanced Peri-LN Transformer scaled with the μP framework, allowing consistent hyperparameter transfer from small to large scales without extensive grid search.

During pre-training, A three-stage curriculum gradually increases the context window, culminating in 128k tokens, which enables THINK to process long documents and perform multi-step reasoning within a single pass. Then, for post-training, we combine supervised fine-tuning on carefully

designed reasoning tasks with Reinforcement Learning from Verifiable Rewards. This alignment strategy encourages the model to generate explicit chains of thought when requested and concise answers when brevity is preferred. Safety alignment follows NAVER AI Ethics guidelines through filtered data, red-teaming, refusal sampling, and policy tuning.

We evaluate THINK on Korea-focused benchmarks such as KMMLU, CSAT, KoBALT-700, HAERAE-1.0, and KoBigBench. The model achieves competitive accuracy among similarly sized models while requiring substantially lower training compute. A vision-augmented variant that integrates vision encoders to extend the same reasoning framework to image-text tasks, matches or surpasses GPT-4.1 on the KCSAT STEM benchmark.

To ensure that academic and industry partners can benefit from the model, we introduce a pruning-and-distillation recipe that reduces parameter count while preserving accuracy. This technique will soon be applied to THINK itself to produce a model suitable for limited resource settings. We plan to open-source release this model under a business-friendly license.

Our contributions are threefold. First, we demonstrate that a regionally tailored corpus combined with modern scaling laws yields a bilingual model with strong reasoning capability. Second, we provide an efficient training and alignment recipe that lowers the barrier to entry for sovereign AI development. Third, we share a practical pruning-distillation pipeline and commit to apply it for an open-source version of THINK—fostering further research and commercial deployment, even under more resource-constrained settings.

2 Pre-Training

This section outlines the pre-training methodology behind THINK: a scalable, Korean-centric data pipeline enriched with targeted synthetic corpora (Section 2.1); a compute–memory-efficient yet stability-oriented Transformer, instantiated with scale-invariant parameterization principles (Section 2.2); and a three-stage curriculum that sequentially builds foundational linguistic knowledge, refines competence with higher-fidelity data, and expands contextual capacity to support long-form reasoning (Section 2.3). See Figure 1 for an overview of the pre-training process.

2.1 Data Preparation

We begin with the end-to-end data pipeline—collection, cleaning, and quality filtering—paying special attention to techniques tailored for our large-scale Korean corpus. We then describe a synthetic-data generation strategy that enriches under-represented domains while preserving linguistic fidelity.

Data Pipeline. The data pipeline for THINK is designed around three guiding principles: scalability, reusability, and quick refresh, so that new corpora can be incorporated with minimal latency while maintaining strict quality guarantees. Following Weber et al. (2024a), the pipeline separates schema standardization from quality assessment and filtering. During standardization, raw documents in heterogeneous formats undergo lightweight cleansing, canonicalization of field names, and storage in a unified schema. The subsequent annotation stage attaches quantitative quality signals, including structural and linguistic metrics, and applies masking to all personally identifiable information (PII). The filtering stage then materializes stage-specific corpora by applying threshold rules to the annotated data and serializes the result into shard files optimized for streaming.

Data Filtering. Korean-specific data filtering schemes have been largely underexplored from the literature. To obtain a corpus that is simultaneously broad and reliably high-quality, we devise a two-tier filtering framework tailored to the linguistic and typographic characteristics of Korean. The first tier extends the rule sets of Weber et al. (2024b) and Lozhkov et al. (2024) by redesigning every heuristic for Korean morphology. Among various quantitative signals, five representative examples—symbol-to-word ratio, mean word length, sentence count, masked-PII ratio, and the proportion of normalized to raw length—are computed for each document. Target ranges for these signals are established through manual inspection with an internal reviewer, and thresholds are further adapted to each source domain (e.g., blogs, wikis) to suppress noise while preserving recall.

The second tier employs model-based scoring. FastText (Joulin et al., 2017, 2016) and transformer encoders are trained under two supervision regimes. In the binary regime, wiki-like passages constitute positive examples whereas noisy web pages form the negative class; the posterior probability

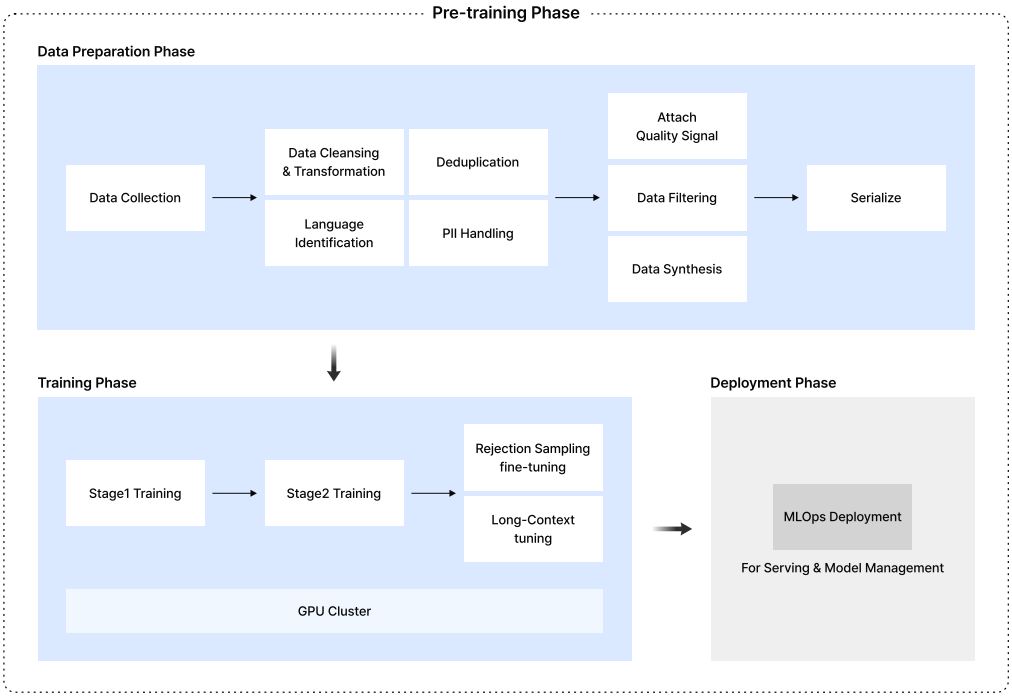


Figure 1: Pre-training pipeline of HyperCLOVA X THINK. (1) Data-Preparation Phase: A scalable pipeline collects raw corpora, carries out cleansing, language identification, deduplication, and masking; attaches quantitative quality signals, applies filtering, synthesizes targeted data, and serializes the resulting shards (2) Training Phase: A dedicated three-stage curriculum, with each stage optimized for its specific objective, progressively builds and refines the model's capabilities.

furnishes a continuous quality score. In the ordinal regime, a language model assigns 0–5 ratings for educational utility, informativeness, and narrative coherence, producing "wiki-like", "educational", and "explanatory" quality predictors analogous to GPT-3, FineWeb-edu, and DCLM filters (Brown et al., 2020; Penedo et al., 2024; Li et al., 2024). A document is retained only if it satisfies a stage-specific conjunction of heuristic thresholds and model scores. Near-duplicates are removed with a MinHash index that is rebuilt at every refresh.

Table 1 summarizes the document-level yield rates of sub-sampled data achieved by the two-tier pipeline. Even within this modest slice, the first stage discards roughly 90 % of raw pages overall, while the more selective second stage retains just 1 – 20 %. These figures reveal aggressive corpus compression, with the pipeline condensing the raw crawl by roughly one to two orders of magnitude even on the sub-sampled slice.

Synthetic Data Generation. In contrast to the extensive curated resources available for major languages (e.g., English and Chinese), high-quality Korean corpus remains markedly underrepresented. To redress this asymmetry, we initiate a systematic program of high-fidelity synthetic data generation, focusing on domains—such as education, law, historical facts, and cultural sentiment—where native Korean content is especially sparse (Yuan et al., 2023; Lee et al., 2024). Leveraging our in-house model family, the pipeline follows two complementary tracks, rewriting existing documents and generating new text from curated seed prompts, while placing filtering and verification at the core of the process to ensure that only high-fidelity Korean data is retained.

The synthetic-data workflow comprises four coupled phases (Cheng et al., 2024; Li et al., 2023; Ben Allal et al., 2024; Su et al., 2024a). (1) Data-design phase: We draft a specification that fixes the target domain, desired volume, file format, and downstream use case. This document governs every subsequent decision in the pipeline. (2) Seed-acquisition and generation phase: License-compliant seed material is collected from open-source and internal repositories. These seeds are either paraphrased to remove copyright artifacts or expanded into new passages through prompt-based genera-

Data	Stage 1 Yield (Filtered / Raw)	Stage 2 Yield (Filtered / Raw)
Total	9.59%	1.36%
Cafe	57.74%	19.84%
Blog	31.53%	2.35%
Web	4.49%	0.27%

Table 1: Stage-wise document yield rates after two-tier filtering.

tion with our in-house language-model family. (3) Filtering and refinement phase: The resulting text is processed by the same two-tier filtering stack used for web data, augmented by routines that detect repetitive templates, logical inconsistencies, and machine-like phrasing. (4) Integration phase: Only data that satisfy all quality checks are versioned and merged into the pre-training corpus, ensuring that synthetic examples extend coverage without degrading overall corpus fidelity. We provide illustrative synthetic data examples in Appendix E. These synthetic corpora are injected into both Stage 1 and Stage 2 of the pre-training curriculum.

2.2 Model Architecture

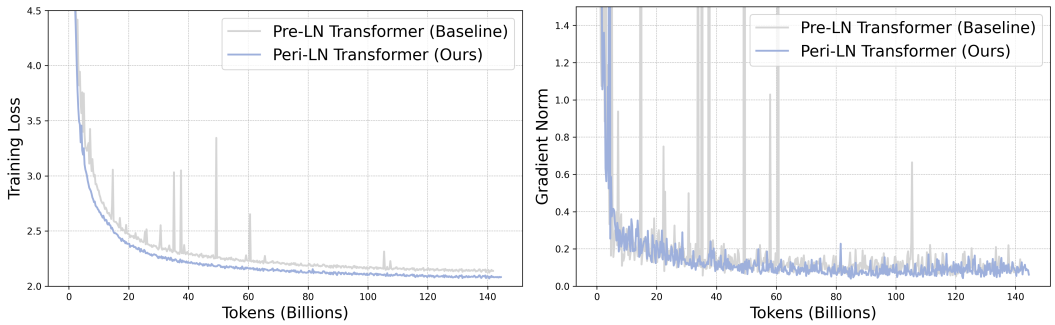
On the architectural front, our design integrates three key components—(i) a compute–memory-balanced Transformer layout (Hoffmann et al., 2022; Rivière et al., 2024), (ii) Peri-Layer Normalization (Peri-LN, Kim et al. (2025)) for training stability and performance, and (iii) Maximal Update Parametrization (μP, Yang and Hu (2021); Yang et al. (2024)) for scale-robust hyper-parameter transfer—together enabling stable scaling and cost-efficient training.

Compute–Memory Balanced Architecture. To minimize compute-bound training cost and memory-bound inference latency under a fixed parameter budget, we employ a *shallower-but-wider* Transformer configuration Hoffmann et al. (2022); Rivière et al. (2024). The model reduces the number of blocks and reallocates the freed parameters to larger hidden and feed-forward dimensions. Because each self-attention layer incurs $O(L^2)$ FLOPs and $O(L)$ activation memory with respect to sequence length L, lowering depth proportionally decreases attention overhead, while widening the FFN, whose cost grows linearly in L, maintains representational capacity.

To empirically substantiate this design, we start from a 3B-parameter baseline comprising 26 Transformer blocks with an FFN hidden size of $7,168$ and generate a shallow-but-wide variant by reducing the depth to 18 layers (30 % shallower) while proportionally increasing the FFN hidden dimension to $11,264$ (57 % wider), thereby conserving the total parameter budget. Owing to the quadratic attention cost, this reallocation lowers the theoretical compute for an 8K-token sequence by 13.7 % TFLOPs. Consistently with this analysis, the modified model ingested 15 % more training tokens within an identical wall-clock budget and matched the validation perplexity of the deeper control, confirming that width-centric capacity reallocation preserves modeling quality while conferring tangible hardware efficiency.

Stability-Oriented Transformer. We stabilize scale-up by coupling Maximal Update Parametrization (μP) with a Peri-Layer-Normalized Transformer. Following the μTransfer procedure, we sweep learning-rate and regularization only on small proxy models, then zero-shot port the optimal settings to each production scale. Because μP preserves update magnitudes across configurations, the large models inherit well-conditioned gradient norms without further tuning, greatly reducing exploration cost while keeping feature learning intact (Yang and Hu, 2021; Yang et al., 2024).

Peri-Layer Normalization (Peri-LN) normalizes both the input and output of every Transformer sublayer, bounding hidden-state variance to grow at most linearly with depth and that layer-wise gradient norms remain stable throughout training.By tightly bounding hidden state statistics, Peri-LN suppresses the massive activations typically observed in Pre-LN models (Sun et al., 2024). Peri-LN also removes the need for FLOP-intensive ablation studies to stabilize architectural or training hyper-parameters. Empirically, Peri-LN yields lower pre-training loss and smaller run-to-run variance (Kim et al., 2025). Maximal Update Parametrization (μP) complements Peri-LN by preserving optimization statistics across width and depth, so hyper-parameters tuned on sub-billion-parameter



(a) Training loss

(b) Gradient-norm

Figure 2: Performance comparison between 8 B-parameter Pre-LN and Peri-LN Transformers during pre-training. Each model size excludes the embedding parameters.

proxies transfer reliably to multi-billion-parameter instances. Together, Peri-LN and μP provide a principled, cost-effective pathway to stable scaling.

To evaluate normalization choices at production scale, we trained two Llama-style models (Dubey et al., 2024) with 8 B parameters on the same open-corpus dataset (Su et al., 2024a), along with our in-house version of the TikToken tokenizer[1]: a standard Pre-LN (Xiong et al., 2020) baseline and an otherwise identical Peri-LN variant. As illustrated in Figure 2, the Peri-LN model exhibits fewer gradient and loss spikes than its Pre-LN counterpart, reproducing the large-scale stability benefits reported by Kim et al. (2025). Furthermore, the Peri-LN configuration attains, on average, a 15 % lower training loss within the same wall-clock budget. These findings confirm that Peri-LN delivers superior stability and performance without incurring additional computational cost, and thus we adopt it as the default normalization scheme in the THINK architecture.

2.3 Pre-Training Curriculum

We adopt a three-staged pre-training curriculum, with each phase focused on a distinct capability target (OLMo et al., 2025; Hu et al., 2024). Stage 1 establishes a general-purpose foundational knowledge base. Stage 2 refines domain-specialized competence by continuing training on high-quality corpora. Stage 3 extends the context window to 128K tokens and internalizes long chain-of-thought reasoning by fine-tuning on rejection-sampled traces generated from an in-house model family. The staged curriculum strategically allocates computational FLOPs to phases with the highest marginal utility, optimizing cost-efficiency while maximizing incremental performance gains.

Stage 1: Foundational Knowledge Construction. The first training stage establishes a broad knowledge base spanning multiple domains. We curate a multilingual corpus, principally Korean and English. Training proceeds on sequences up to 8K tokens, consuming 6 trillion tokens in total. The learning rate is linearly increased during the initial $5,000$ steps to a peak of $1.59e$-3 determined by μP scaling, after which it is annealed according to a cosine schedule to $1.59e$-4 (10 % of the maximum), thereby promoting stable convergence.

Stage 2: Domain-Specialized Capability Boosting. The mid-training stage introduces an additional 1 trillion tokens to sharpen the model's domain expertise and reasoning ability while maintaining the 8K-token context length established in Stage 1. We gradually down-weight generic web text and increase high-quality, domain-focused corpora including the synthetic datasets constructed in Section 2.1. A brief $2,000$-step warm-up ensures a smooth transition to these revised distribution.

Guided by Bi et al. (2024), for learning rate schedule, we adopt a two-step decay profile: the rate is held at $1.59e$-4 for 80 % of training, reduced to 31.6 % of this peak ($\approx 4.76e$-5) for the next 10 %, and finally to 10 % ($\approx 1.59e$-5) for the last 10 %. For the data mix, following Blakeney et al. (2024), we rebalancing the dataset during the final 10 % of training steps. Sampling of lower-quality general text is gradually reduced. Conversely, the sampling weight of under-represented domains, crucial

[1] https://github.com/openai/tiktoken

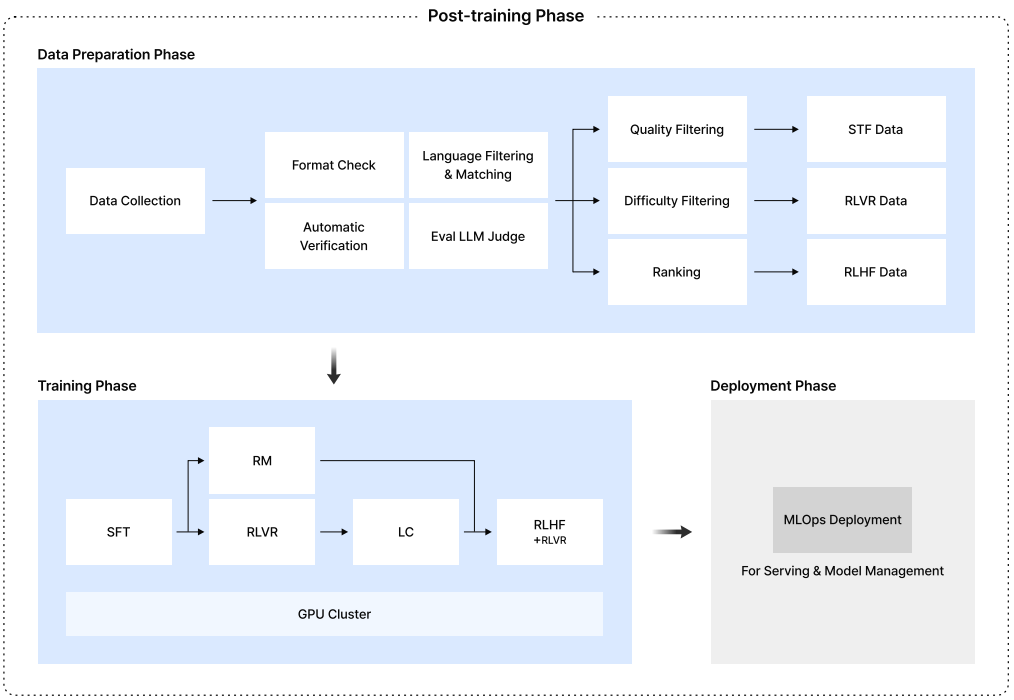


Figure 3: Post-training pipeline of HyperCLOVA X THINK. (1) Data-Preparation Phase: Data is collected and then rigorously processed through steps such as format validation, automatic verification, language-based filtering, and evaluation by LLM-based judges. The data is refined through quality filtering, difficulty filtering, and ranking to prepare data suitable for subsequent training stages with different objectives. (2) Training Phase: A sequence of fine-tuning procedures—including Supervised Fine-Tuning (SFT), Reward Modeling (RM), Reinforcement Learning with Verifiable Rewards (RLVR), training for reasoning Length Controllability (LC), and Reinforcement Learning from Human Feedback (RLHF) —is executed across a large-scale GPU cluster.

for sovereign-AI applications, is increased, with emphasis on Korean medical literature, national economic reports, and culturally contextualized historical archives.

Stage 3: Extended Context Alignment. Standard corpora are biased toward short documents; naively over-sampling longer texts therefore disrupts training stability (Zhuang et al., 2025). We mitigate this issue with *length-based, proportion-preserving resampling*, which increases the number of long documents while maintaining each length bucket's share of total tokens. After pre-training with an 8 K context window and a rotary-position-embedding base θ of 500 K, we expand the window in three successive stages—32 K, 64 K, and 128 K. At each expansion, θ is raised from 500 K to 5 M, then to 20 M, and finally to 100 M. A brief warm-up followed by cosine decay restores perplexity before the next enlargement (Su et al., 2024b; Xu et al., 2024). To supply explicit supervision for extended reasoning, we additionally train on a long chain-of-thought corpus generated in-house and filtered via rejection sampling (Yuan et al., 2023; Lee et al., 2024) (see §2.1). This synthetic dataset spans up to 128 K tokens, enabling the model to master long-context conditioning without degrading the general or domain-specific competencies obtained in Stages 1 and 2.

3 Post-Training

This section outlines the post-training methodology of THINK: a supervised fine-tuning (SFT) phase that injects core reasoning patterns and task-specific capabilities (Section 3.1); and a multi-stage reinforcement learning pipeline that incorporates verifiable rewards, length-controllability, and human feedback to achieve aligned, efficient, and scalable reasoning (Sections 3.2–3.4). See Figure 3 for an overview of the training process.

Reasoning Mode	Non-Reasoning Mode
<|im_start|>user {query}<|im_end|> <|im_start|>assistant/think {reasoning}<|im_end|> <|im_start|>assistant {response}<|im_end|><|endofturn|>	<|im_start|>user {query}<|im_end|> <|im_start|>assistant {response}<|im_end|><|endofturn|>

Table 2: Unified chat template used for training models to support both reasoning and non-reasoning interaction modes.

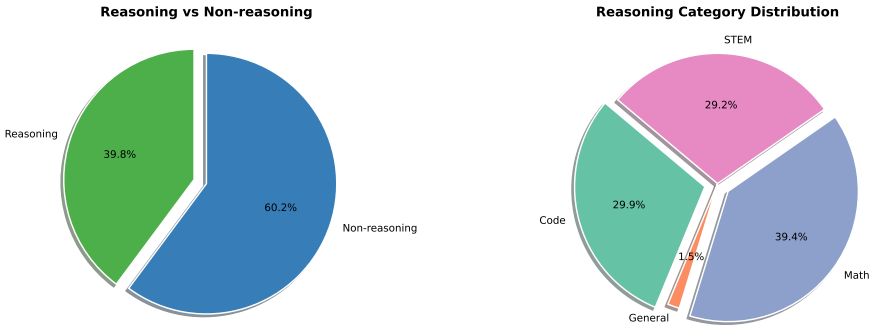


Figure 4: Data distribution utilized for Supervised Fine-Tuning (SFT), reflecting a balanced composition tailored to support effective downstream reinforcement learning and reasoning capabilities.

THINK is trained to operate in an integrated manner, allowing for dynamic switching between a detailed 'reasoning mode' for complex, multi-step reasoning and a more direct 'non-reasoning mode' for rapid, context-driven responses. This unified framework eliminates the need for users to switch between separate models (e.g., a dedicated reasoning model and a chatbot), as illustrated in Table 2.

3.1 Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning (SFT) serves as a foundational step in our post-training pipeline, aiming to inject desired behaviors and reasoning patterns into the model. This stage establishes a strong base for subsequent reinforcement learning phases.

The dataset used for SFT is constructed by aggregating various sources across mathematics, coding, STEM, and general abilities. We carefully curate data from a series of ablation studies and utilize high-quality open-source and in-house data. For reasoning data, each sample contains prompt, assistant think, and assistant response. The assistant think contains a rather free-form chain-of-reasoning, while the assistant response is a concise, finalized output that directly answers the user's query based on that reasoning. The general statistics for the SFT dataset is illustrated in Figure 4.

To ensure data quality and consistency, we apply a multi-stage filtering pipeline across all datasets. Each item in data goes through a basic format check to ensure that the output contains proper format (e.g., boxed answers for math problems and compilability for code problems). Language filtering is applied to select only samples written in the target language, and language matching further ensures that input and output languages are the same for each sample. For reasoning data specifically, we also check whether the final answer is automatically verifiable. For non-reasoning data, we employ a LLM-as-a-Judge method to score each example by their helpfulness and safety and filter out those with low scores.

Training is performed with dynamic batching to fill each batch dynamically to its maximum capability, in order to optimize GPU utilization and memory usage. The model is trained over 4 epochs

with early stopping based on validation accuracy. Similarly to other reports (Yang et al., 2025), we observe that selecting a checkpoint from later epochs results in reduced exploration of the model during the subsequent phase. More comprehensive details on the training setup of SFT are provided in our previous technical report (Yoo et al., 2024b).

3.2 Reinforcement Learning with Verifiable Rewards (RLVR)

The Reinforcement Learning with Verifiable Rewards (RLVR) is crucial for improving reasoning capabilities through verifiable feedback mechanisms. The main objective is to optimize model performance by accurately guiding behavior through precise rewards and penalties.

Reinforcement Learning Algorithm. In our implementation of RLVR, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

Unlike more traditional RL algorithms, it calculates a baseline advantage based on multiple generations per prompt, optimizing computational efficiency and maintaining training effectiveness. To further enhance the robustness and accuracy of our RLVR framework, we introduce several targeted modifications:

- **KL Divergence Penalty Removal:** Our initial experiments indicated that this penalty restricts models from exploring diverse behaviors and incurs significant computational overhead due to the necessity of inference from a reference model. Removing the penalty improved computational efficiency and model flexibility.
- **Constant Normalization:** We observed that prompt difficulty often correlates with response length–more difficult prompts tend to produce longer responses–thereby introducing biases related to response length. To mitigate these biases from varying response lengths and prompt difficulties, we adopt constant normalization strategy from Liu et al. (2025).
- **Relaxed Upper Bound for Exploration:** To encourage exploration and prevent deterministic policy collapse, we adopt the clip-higher approach (Yu et al., 2025), which raises the upper threshold of the importance sampling ratio in GRPO. By including low-probability tokens into policy updates, this approach increases policy entropy and fosters diverse reasoning paths.

Collectively, these methodological enhancements enable our RL training to achieve an optimal balance of exploration, computational efficiency, and stable training performance.

Data Efficiency. To optimize training efficiency, enhance model performance, and effectively utilize computational resources, we employ targeted difficulty filtering techniques, including both offline and online methods.

We implement offline difficulty filtering to our dataset by excluding prompts that are either too easy or too challenging. Specifically, we leverage predictions generated by the SFT checkpoint—our initial model for RLVR—to evaluate the difficulty of each prompt. By sampling multiple responses from this checkpoint, we calculate the average accuracy of predictions and remove prompts with accuracy of exactly 0.0 or 1.0. This strategy ensures the inclusion of prompts only with appropriate difficulty levels at the outset of training.

However, offline difficulty filtering has limitations. Because this filtering method occurs only once before the training begins, it is inherently static. As the model's performance improves as the training progresses, the dataset's difficulty level cannot be adjusted accordingly–a problem that was once challenging can become solvable. Consequently, this static nature can lead to discrepancies between evolving model capabilities and fixed difficulty of prompts.

To address the shortcomings of offline filtering, we additionally incorporate an online difficulty filtering strategy. Utilizing GRPO allows us to generate multiple responses per prompt within each batch. For each group, we calculate accuracy and remove prompts where all generated responses are either entirely correct or entirely incorrect from the batch. This dynamic filtering approach continuously adapts the training set's difficulty to the model's evolving capabilities, ensuring that learning remains focused on informative examples and thereby maintaining optimal training efficiency.

Our analysis aligns with recent findings suggesting that online difficulty filtering effectively optimizes the lower bound learnability of reinforcement learning algorithms by dynamically balancing

prompt difficulty (Bae et al., 2025). Importantly, we observe that even with initial offline filtering, online filtering still provides substantial additional benefits. Thus, combining both offline and online difficulty filtering significantly enhances our training efficiency and model performance.

Reward Shaping. To effectively guide model training and enhance its performance in our RLVR framework, we carefully design a reward shaping strategy consisting of several distinct components:

- **Format Reward:** We establish a set of format rules that responses must follow. To calculate this reward, we count the number of rules adhered to by the model's response and divide it by the total number of format rules. We adopt this soft penalty approach as it demonstrates minimal negative impact on reasoning performance, allowing models to progressively align with the desired response structure without detrimental effects.
- **Language Reward:** This reward is computed based on the ratio of characters generated in the same language as the prompt. By directly correlating the language of responses with the language of prompts, this reward encourages the model to reason in the intended language, significantly enhancing multilingual reasoning capabilities.
- **Verifiable Reward:** We incorporate verifiable rewards across multiple problem categories, including mathematics, code generation, code input-output (Code IO), and multiple-choice questions. The verification outcomes directly determine reward allocation, with a binary value: a fully correct response receives a reward of 1.0, while any incorrect response results in a reward of 0.0.
- **Overlong Reward:** We adopt both Soft Overlong Penalty and Overlong Loss Masking (Yu et al., 2025), because penalizing truncated samples harshly can introduce undesirable reward noise, potentially destabilizing training by penalizing valid reasoning solely due to length. The former gradually increases as the response length exceeds the predefined maximum value, and the latter masks the loss of truncated samples, effectively stabilizing the training process.

Optimized Rollout Sampling Process. Efficiency in the rollout sampling process is crucial for optimizing the RLVR training pipeline, as this stage typically dominates the overall training duration. To address this, we implement a highly efficient asynchronous sampling procedure. In this setup, inference nodes are utilized continuously and concurrently until the number of completed rollout samples meets or exceeds the training batch size. Samples generated from these inference nodes are collected and stacked asynchronously, significantly reducing idle times and improving resource utilization.

Moreover, due to our implementation of online difficulty filtering, certain samples may be dynamically filtered out during the rollout process, potentially causing delays or inefficiencies. To counteract this, we maintain a buffered approach to concurrent sampling, ensuring multiple samples are processed simultaneously. This strategy effectively compensates for any filtered-out examples by ensuring continuous generation of alternative samples, thereby minimizing or entirely masking the time loss associated with discarded examples. This optimized asynchronous sampling approach greatly enhances the efficiency and stability of the RLVR training process (Bae et al., 2025).

3.3 Reasoning Length Controllability (LC)

Reinforcement learning with Large Reasoning Models (LRMs) enables drastic improvements in complex reasoning capabilities, but often accompanies undesired tendencies to overthink (Chen et al., 2024; Sui et al., 2025) or even underthink (Wang et al., 2025) compared to the optimal reasoning length. For practical and flexible deployment of computationally expensive LRMs, we identify *length controllability* (LC) as a key desideratum. To induce LC in HyperCLOVA X THINK, we additionally incorporate the length-penalized reward functions introduced by Aggarwal and Welleck, 2025.

On top of the training configurations from the previous RLVR stage, we train our model on the length-penalized reward functions (L1-Exact and L1-Max) from Aggarwal and Welleck, 2025. We append '`Think for maximum N tokens`' on the input instructions, where we sample N from a discrete token budget set of $\mathcal{B} = \{1024, 2048, 4096, 8192, 16384\}$ to accelerate LC capability[2].

[2]The original L1 paper randomly samples N from $\mathcal{U}_{[100,4000]}$

We first train the model on the L1-Exact penalty for about 300 steps to acquire LC and subsequently about 100 steps on the L1-Max penalty to greedily reduce the reasoning length when possible.

3.4 Reinforcement Learning from Human Feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF) aligns model outputs with human preferences and practical usability. By combining reasoning/non-reasoning RLHF and RLVR, we concurrently refine model behavior to improve alignment with human preferences while preserving and enhancing reasoning abilities.

To better align the model's outputs with human preferences, we first train a reward model using a combined set of human preference data, as detailed in our previous technical report (Yoo et al., 2024b). This data consists of pairwise comparisons either annotated by expert raters or inferred via scoring from in-house judge models. The reward model learns to predict scores for each sequence in non-reasoning data. Following this, we use GRPO explained in Section 3.2 as the core RLHF algorithm. The policy is optimized to maximize the expected reward predicted by the reward model. Unlike RLVR, we apply a KL penalty of 0.1 to maintain proximity to the SFT checkpoint. This relatively strong KL penalty prioritizes training stability over exploration in RLHF.

The prompts used during RLHF training consist of a mixture of reasoning and non-reasoning tasks. For non-reasoning, the model is expected to generate assistant response directly, while for reasoning, the model first generates intermediate think step followed by assistant response. The reward model evaluates only the response portion of the output and the think portion is not directly scored, allowing the model to freely develop internal reasoning patterns.

Lastly, when training with RLHF subsequently after RLVR, we observe a slight degradation in the model's reasoning ability that was optimized during the RLVR phase. A similar pattern was also observed in other reasoning models (Yang et al., 2025). To address this issue, we adopt a joint training strategy where RLVR and RLHF are trained concurrently. Specifically, we interleave the training batches such that each batch contains a mixture of samples from RLVR and RLHF datasets. This approach preserves the performance gains of both RLHF and RLVR while unifying the training phases, resulting in a simpler and more effective training pipeline.

4 Evaluation

4.1 Baselines

We compare our model against publicly available models of comparable size that are recognized for their reasoning capabilities, including Qwen3-14B, Qwen3-32B (Yang et al., 2025), QwQ-32B (Qwen Team, 2025), and EXAONE-Deep-32B (LG AI Research, 2025). We utilize evaluation scores directly from each model's original paper when available. Otherwise, we conduct our own evaluations and report the corresponding results.

4.2 Evaluation Protocol

When published metrics are unavailable, we perform in-house evaluations using primarily public benchmark sets, with the exception of KoBigBench (Yoo et al., 2024b). The primary goal of our evaluation strategy is to interpret and extract the predicted answers from language models for both open-ended and multiple-choice benchmarks as accurately as possible. Models often fail to produce a final answer when asked to generate the reasoning chain and the answer consecutively in a single pass. To address this, we adopt a two-pass generation scheme: the model first produces the reasoning chain with our chat template (`<|im_start|>assistant/think\n...<|im_end|>`), and we then generate the answer by appending an answer prefix. Our evaluation framework combines LM Eval Harness Gao et al. (2023) with an in-house toolkit that we plan to release soon for public reference.

For our model, we configure the generation temperature at 0.5 and top-p at 0.95. In the case of other models, their authors' recommended optimal hyperparameters are utilized. All evaluations are performed using zero-shot Chain-of-Thought (CoT) reasoning, and the maximum CoT generation length is uniformly set to 4096.

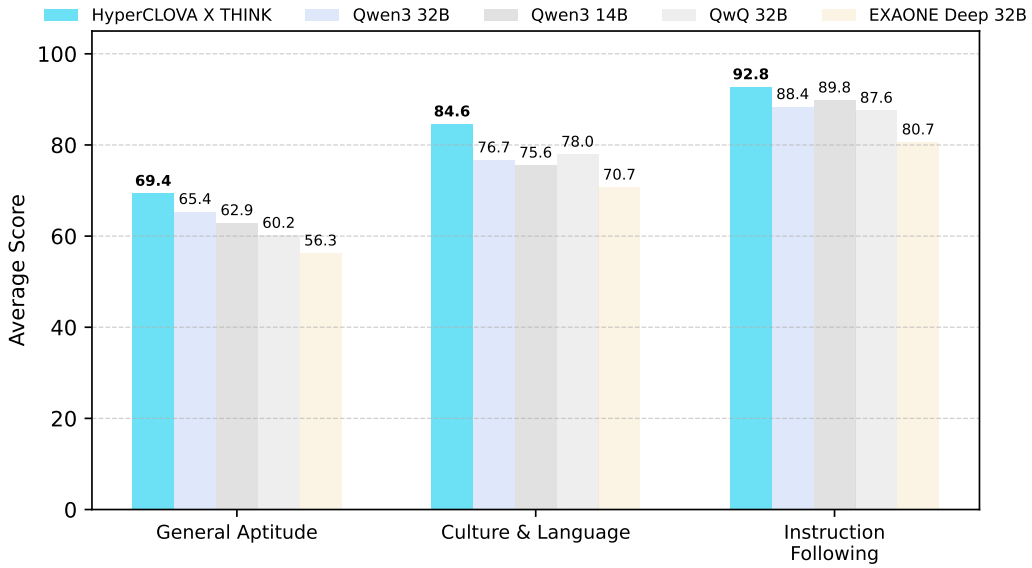


Figure 5: Summary of model performance on (1) General Aptitude, (2) Culture and Language, and (3) Instruction-following benchmarks for the Korean domain. The instruction-following benchmark scores are normalized by multiplying their original values by 10.

4.3 Korea-Centric Benchmarks

Setup. As introduced in Section 1, our model's general performance is evaluated against various baselines using a set of Korea-centric benchmarks. These evaluations are designed to assess the model's understanding of Korean culture and knowledge. To achieve this, we curated datasets specifically related to the Korean domain:

- **General Aptitude:** KMMLU (Son et al., 2025) and CSAT gauge general Korean knowledge. KorMedMCQA (Kweon et al., 2024) focuses on medical problem-solving and KoBALT-700 (Shin et al., 2025) assesses linguistic depth and typological grounding in Korean.
- **Culture and Language:** HAERAE-1.0 (Son et al., 2024), CLIcK (Kim et al., 2024a), and KoBigBench[3] evaluate Korean-specific cultural, geographical, historical knowledge, etc.
- **Instruction-Following:** LogicKor (Park, 2024) and KoMTBench (LG AI Research, 2024) measure the model's ability to follow Korean instructions.

Result. Our model's strong performance on the comprehensive aptittude tests, Korean-specific culture and linguistic benchmarks, and a suite of benchmarks for probing instruction-following capabilities is summarized in Figure 5 and detailed in Table 3. By employing zero-shot CoT prompting to elicit robust reasoning and evaluating answers based on accuracy, we demonstrate that THINK surpasses other baselines. Additional evaluation results can be found in the Appendix B and Appendix C. Furthermore, this superior performance is achieved with a relatively small computational cost, which will be discussed further in the subsequent section.

5 Analysis

5.1 Training Efficiency

There has been research showing that model performance consistently improves with increases in data volume, parameter count, and computational resources in accordance with Scaling Laws (Kaplan et al., 2020). This has also been further supported by more recent work on Expanded Neural

[3]The dataset will be publicly released.

Category	Benchmarks	HCX THINK (-)	Qwen3 (32B)	Qwen3 (14B)	QwQ (32B)	EXAONE Deep (32B)
General Aptitude	**KMMLU**	**69.7**	63.5	49.3	54.1	53.6
	CSAT	83.2	81.9	77.1	**84.7**	69.7
	KorMedMCQA	**76.0**	74.7	68.5	69.4	68.8
	KoBALT	**48.9**	41.4	38.4	32.4	33.0
Culture & Language	**HAERAE**	**87.8**	75.1	74.1	76.2	74.7
	CLIcK	**80.1**	71.1	68.8	73.6	62.2
	KoBigBench	**85.9**	83.9	83.8	84.1	75.3
Instruction-Following	**LogicKor**	**9.65**	8.93	9.15	9.02	8.54
	KoMTBench	**8.90**	8.75	8.82	8.50	7.59

Table 3: Performance comparison of language models on Korea-centric benchmarks. Models are evaluated across comprehensive understanding, cultural sovereignty, and chat-based instruction-following tasks, highlighting their capabilities and adaptability within a Korean context.

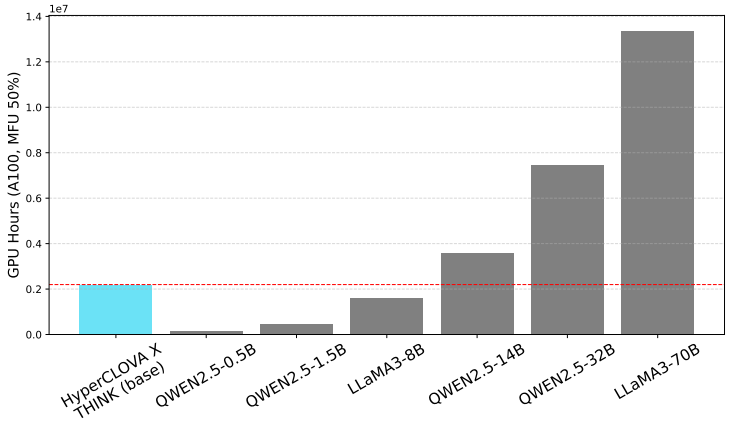


Figure 6: Training Efficiency (GPU Hours / A100 / MFU 50%)

Scaling(Chang et al., 2024). However, recent studies have increasingly emphasized the importance of data quality. For example, Chang et al. (2024) quantifies data diversity and quality through the concept of effective training tokens and proposes a corresponding scaling law. This study experimentally demonstrates that efficient training and performance improvements are achievable even for smaller models.

It was reported that Qwen2.5 significantly improved its reasoning and long-context generation capabilities using 18 trillion tokens of high-quality training data and advanced post-training strategies(Qwen et al., 2025). Similarly, LLaMA 3, trained on 15 trillion tokens, showed continued performance gains even after surpassing the Chinchilla-optimal range. This suggests that while data scaling remains important, an approach centered on data quality is also necessary. Furthermore, as the volume of natural language data available for collection from the internet approaches its limits, a paradigm shift from data quantity to data quality is accelerating.

THINK is developed with a focus on creating a high-efficiency architecture and a training strategy grounded in high-quality data. As a result, it required significantly fewer GPU hours than similar sized models to be trained, as shown in Figure 6. At the same time, it achieves competitive perfor-

	Cross-lingual Consistency (En, Ko)				MT	
	(✓, ✓) ↑	(✓, ✗) ↓	(✗, ✓) ↓	(✗, ✗) ↑	Ko→En	En→Ko
Qwen3 32B	**81.0**	**8.0**	**4.5**	6.5	**92.8**	85.3
EXAONE Deep 32B	62.5	22	3.5	**12.0**	85.6	77.5
HyperCLOVA X THINK	74.5	12.0	**4.5**	9.0	90.3	**85.8**

Table 4: Cross-lingual transferability between English and Korean. Each consistency column shows the case of MCQA items for which the model is correct (✓) or incorrect (✗) in English (first symbol) and Korean (second symbol). Higher symmetric ((✓, ✓) and (✗, ✗)) and lower asymmetric ((✗, ✓), (✓, ✗)) ratios imply stronger consistency. We also report xCOMET translation quality of Flores on both directions.

mance. This demonstrates that strategic data curation and training efficiency are critical factors in developing high-performance LLMs, moving beyond reliance on sheer resource input.

5.2 Cross-Lingual Transferability

In this subsection, we investigate cross-lingual consistency and bi-directional translation quality between Korean and English to evaluate whether the model properly transfers the acquired English knowledge into Korean. Our cross-lingual evaluation hypothesis is twofold. First, a model that has efficiently encoded both languages should yield semantically equivalent answers when parallel questions are posed in English and Korean. Second, if the same underlying representations truly capture language-agnostic meaning, the model should also display strong translation ability in both directions.

Cross-lingual Consistency. In order to compute the score of cross-lingual consistency, we adopt the pipeline proposed by Qi et al. (2023); Xing et al. (2024); Yoo et al. (2024a) and evaluate with Global-MMLU-Lite (Singh et al., 2024). We only computed the scores in culturally agnostic samples to exclude examples whose gold answers depend on the source language. Our experiment utilizes Caliper framework, described in Section. 4.2. We categorize the model's predictions on parallel English-Korean MCQA prediction results into four cases: (1) (✓ ,✓) represents question answered correctly in both languages indicating the desired cross-lingual aligned, (2) (✓, ✗) is the number of samples that are correct in English but incorrect in Korean, isolating failures of knowledge transfer, (3) (✗, ✓) represents the opposite scenario, and (4) (✗, ✗) records items answered incorrectly in both languages, reflecting residual knowledge gaps. This decomposition enables us to attribute improvements in overall accuracy to genuine bilingual robustness.

As shown in Table 4, THINK achieves a comparable (✓,✓) ratio (74.5%) ,only a few points behind the extensively trained Qwen3 32B, while limiting asymmetric errors to 16.5%. Given that THINK was tuned almost exclusively on the Korean–English pair and required a fraction of the compute budget demonstrated in Section 5.1, this result indicates that carefully targeted bilingual training can offset much of the scale advantage by large multilingual models. Although the remaining asymmetric cases highlight room for improvement, THINK already delivers a robust and cost-efficient bilingual representation. Its overall consistency is also higher than that of EXAONE Deep 32B, suggesting that strategic data curation can sometimes outweigh pure parameter count.

Machine translation. To complement the cross-lingual transferability analysis in cross-lingual consistency with MCQA task, we next assess bidirectional machine translation (MT) performance of each model between Korean and English. We adopt the Flores benchmark Team et al. (2022) and translate the official test split in both directions (En→Ko, Ko→En). A prompt of each model includes the same 1-shot example. We only extract the translation part from the response. Each translation quality is measured with xCOMET-XL Guerreiro et al. (2023), a model based metric that has a stronger correlation with professional human judgment compared to BLEU and ChrF. We report xCOMET-XL score for each direction. This performance indicates the model can faithfully re-express the same underlying knowledge as fluent Korean or English.

MT columns of Table 4 provide additional evidence of THINK's robust cross-lingual transferability in a generation task. In Ko→En direction, THINK achieves a competitive xCOMET score as 90.3,

closely approaching the performance of Qwen3 32B model (92.8). Furthermore, in the opposite direction (En→Ko), THINK surpasses all other models with a score of 85.8. This result indicates that our training pipeline not only preserves English knowledge but also enhances the model's ability to render it into high-quality Korean. These findings, combined with the consistency results, validate that our data curation can deliver bidirectional translation capability without the extensive computational overhead of full-scale multilingual pre-training.

6 Extensions

6.1 Instilling Vision-Language Reasoning in Korean

The pursuit of sovereign multimodal AI requires not only proficiency in native languages but also robust capabilities for reasoning across modalities. Given that THINK was originally developed and optimized for advanced reasoning in text, can it be effectively extended into vision-grounded reasoning through a dedicated *multimodal post-training pipeline*?

In this subsection, we present a separate experimental branch: Starting from the text SFT pipeline (Section 3.1), we incorporate visual modules and multimodal tuning to construct a vision-language model. This enables direct evaluation of complex vision-language reasoning beyond simple transfer from text-only capabilities. For real-world assessment, we use challenging STEM items from the Korean College Scholastic Ability Test (KCSAT). As the KCSAT is administered in Korean and reflects rigorous local standards, it is suitable as a test of vision-language reasoning ability in Korean.

Each item is presented to the model as an image containing mathematical expressions, tables, diagrams, and scientific text (See Appendix D). The model must first accurately recognize visual content (e.g., text, layout, object and diagram recognition), then perform multi-step logical reasoning. Here, *vision-language reasoning* refers to this integrated process of visual understanding and abstract problem-solving, beyond perceptual recognition alone.

Architecture and Training. For vision-language reasoning, we augment the LLM backbone with a visual encoder module, similar to the architecture in our previous work, HyperCLOVA X SEED (HyperCLOVA X Team, 2024). The architecture is composed of:

- **Vision Encoder:** SigLIP-2 (Tschannen et al., 2025), operating at 512×512 pixels per grid.
- **Vision-Language Model Architecture:** LLaVA-1.5-HD-based framework (Liu et al., 2024) with C-Abstractor (Cha et al., 2024) connector mechanism, supporting up to 1.57M total pixels distributed over 6 grids.

The training pipeline extends our previous protocol (Kim et al., 2024b) by inserting a dedicated vision SFT stage between text SFT and multimodal RLHF. More concretely, after pre-training on large-scale text corpora, we first apply SFT on instruction-oriented text data, then conduct multimodal SFT with paired image-text data, and finally perform multimodal RLHF on both text-only and multimodal instructions. This change to RLHF—incorporating vision-language samples in addition to text—distinguishes our ablation pipeline from standard text-only approaches. Reasoning Mode (Section 3) is toggled via explicit prompting throughout both model training and inference, with ablations performed using both reasoning-enabled and baseline prompt formats.

Experiments. We evaluate vision-language reasoning performance on the multimodal Korean Educational Test benchmark (Park and Kim, 2025), with primary focus on its most difficult subset: the KCSAT STEM subjects. This evaluation set comprises 206 items spanning mathematics, physics, chemistry, earth science, and biology, each requiring advanced logical and visual inference at both basic and advanced levels. The KCSAT is internationally recognized for its depth and rigor, making it an exemplary proxy for high-stakes, real-world STEM reasoning.

Our experiments compare THINK with Vision against leading contemporary closed APIs in the multimodal LLM space—specifically GPT-4 Turbo with Vision (OpenAI et al., 2024c), GPT-4o (OpenAI et al., 2024a), GPT-4.1 (OpenAI et al., 2024c) and OpenAI-o1 (OpenAI et al., 2024b). All models are assessed under strictly identical protocols using the same visual and textual input, ensuring a fully standardized evaluation environment.

Results and Analysis. As summarized in Table 5, THINK attains an overall accuracy of 46.4% on the KCSAT STEM benchmark, outperforming GPT-4.1 (40.3%) and approaching the performance of

Model	Math		Physics		Chemistry		Earth Science		Biology		Overall
	Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	
GPT-4 Turbo with Vision	54.5	20.8	5.0	15.0	15.0	20.0	30.0	25.0	10.0	40.0	23.8
GPT-4o	68.2	50.0	15.0	20.0	25.0	25.0	40.0	30.0	15.0	25.0	32.0
GPT-4.1	68.2	66.7	20.0	30.0	40.0	30.0	30.0	40.0	35.0	35.0	40.3
OpenAI-o1	93.2	83.3	42.5	40.0	38.8	56.3	32.5	42.5	37.5	31.3	50.9
HyperCLOVA X THINK with Vision	68.2	68.1	33.3	28.3	41.7	58.3	28.3	38.3	43.3	50.0	46.4
w/o Reasoning Mode	22.7	20.8	11.7	26.7	11.7	28.3	20.0	23.3	30.0	21.7	21.7

Table 5: Evaluation of native vision-language reasoning ability on the KCSAT STEM multimodal benchmark (Park and Kim, 2025) by subject and level. The benchmark consists of 206 questions covering five scientific subjects (basic/advanced), with 20 to 24 questions per subject and level. KCSAT is widely regarded for its difficulty, emphasis on scientific reasoning, and its reflection of Korea's high-achieving STEM education system.

GPT-o1 (50.9%). Disabling Reasoning Mode notably causes performance to drop to 21.7%, supporting the conclusion that advanced reasoning skills acquired during language pretraining are crucial and can be effectively extended to vision-centric STEM challenges when combined with specialized multimodal tuning. Further qualitative analyses and representative sample outputs are provided in Appendix D.

On the other hand, we note a modest trade-off: adding multimodal SFT introduces a slight decrease in text-only reasoning performance, underscoring the inherent difficulty of jointly optimizing for both modalities. Achieving a balanced sovereign AI—excelling in both unimodal and multimodal reasoning—remains open for further tuning and methodological advances. These observations highlight that effective vision-language reasoning, especially in STEM, demands robust integration of visual parsing and native multi-step logic. As a next step, our future work will extend the model's capabilities towards unified, native reasoning across text, vision, and audio.

6.2 Lightening through Pruning and Distillation

As competition in high-performance model development intensifies, training models with tens or hundreds of billions of parameters using trillions of tokens has become the de facto industry standard. This large-scale training approach entails high costs and long development cycles, while limiting the ability to respond to rapidly changing service environments. Consequently, learning strategies that can efficiently build LLMs with fewer resources have recently gained attention. These approaches not only reduce costs, but also offer practical advantages in terms of timely model development and operation.

One of the leading approaches combines pruning and knowledge distillation. Pruning reduces model size by removing less important parameters, while distillation is a technique that transfers knowledge learned by large models to lightweight models to maintain performance. Combining these two techniques can achieve both model compression and performance preservation simultaneously.

As a real-world example, HyperCLOVA X SEED 0.5B, recently released on HuggingFace, is the first open-source model in the HyperCLOVA X series trained using pruning and knowledge distillation. Despite being similar in size to Qwen2.5-0.5BQwen et al. (2025), it was trained at approximately 39 times lower cost and outperformed competing models in most benchmarks. Notably, it demonstrated significant performance improvements in Korean language benchmarks. This model offers high practical value as it enables high-performance conversational interfaces even in resource-constrained environments such as mobile applications or smart home devices.

Furthermore, the combination of pruning and distillation can be utilized for efficient production of both lightweight and large models. Depending on the structure of the teacher model used for training, the type of data to be transferred, and the learning strategy, models of various sizes and purposes can be produced. This flexibility is expected to improve usability across future generative AI applications.

Currently, a pruned and distilled version of THINK is under preparation to be open-sourced.

7 Conclusion

In this report, we introduced HyperCLOVA X THINK, the first reasoning-focused LLM within HyperCLOVA X family. It is efficiently trained to achieve two primary objectives: advanced reasoning capabilities and the promotion of sovereign AI for Korea.

Its pre-training dataset comprises approximately 6 trillion high-quality tokens spanning Korean, English, and further enhanced by targeted synthetic Korean data. We employ a Peri-LN Transformer scaled with μP, ensuring stable scalability and cost-efficient training. A three-stage curriculum enables the model to expand its context window to 128k tokens and demonstrate robust long-form chain-of-thought reasoning. Post-training involves supervised fine-tuning and reinforcement learning with verifiable rewards, utilizing a curated data filtering process to address both detailed reasoning and simple answering tasks.

Experiments demonstrate HyperCLOVA X THINK's competitive performance against other reasoning models on Korea-centric benchmarks such as KMMLU, CSAT, KoBALT-700, HAERAE-1.0, and KoBigBench. Analysis highlights its highly efficient training cost and its ability to preserve robust bilingual consistency. Furthermore, a vision-augmented variant achieved performance comparable to GPT-4.1 on the KCSAT STEM benchmark.

Our report shows that using additional test-time compute to refine model responses is an effective way to push the limits of model capability and improve compute-cost efficiency. As foundational AI technology gains greater potential to enrich people's lives and shape the future of digital business, we remain committed to improving reasoning scalability and delivering foundation models that are both powerful and affordable, thereby accelerating both domestic and global AI transformation in businesses.

Note that, while we took necessary measures to improve the safety of HyperCLOVA X THINK as per NAVER AI Ethics guidelines, the harmlessness of the generated text cannot be fully guaranteed. Thus, the responses may contain toxic remarks, exhibit biases or otherwise harmful content. However, we remain dedicated to responsible AI development and deployment.

Lastly, we plan to open-source a pruned and distilled version of HyperCLOVA X THINK. This initiative aims to benefit academic and industry partners with limited resources, fostering the future development and utilization of sovereign LLMs.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954.

Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. 2024. Does your data spark joy? performance gains from domain upsampling at the end of training. *CoRR*, abs/2406.03476.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced Projector for Multimodal LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ernie Chang, Matteo Paltenghi, Yang Li, Pin-Jie Lin, Changsheng Zhao, Patrick Huber, Zechun Liu, Rastislav Rabatin, Yangyang Shi, and Vikas Chandra. 2024. Scaling parameter-constrained language models with quality data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 80–97, Miami, Florida, US. Association for Computational Linguistics.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2529–2550. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *arXiv preprint arXiv:2310.10482*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kai Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *CoRR*, abs/2404.06395.

NAVER Cloud HyperCLOVA X Team. 2024. HyperCLOVA X SEED Vision Instruct 3B. `https://huggingface.co/naver-hyperclovax/HyperCLOVAX-SEED-Vision-Instruct-3B`. Available on Hugging Face Hub. Accessed: 2025-06-23.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Eunsu Kim, Juyoung Suk, Philhoon Oh, Haneul Yoo, James Thorne, and Alice Oh. 2024a. CLIcK: A benchmark dataset of cultural and linguistic intelligence in Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3335–3346, Torino, Italia. ELRA and ICCL.

Geewook Kim, Taeho Kil, and Jinbae Im. 2024b. "HyperCLOVA X Vision: Open Your Eyes, CLOVA X!". https://tv.naver.com/v/67447111. Conference Talk, Dan24, Naver AI NOW, December 21, 2023. Accessed: 2025-06-23.

Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. 2025. Peri-ln: Revisiting normalization layer in the transformer architecture. In *Forty-second International Conference on Machine Learning*.

Sunjun Kweon, Byungjin Choi, Gyouk Chu, Junyeong Song, Daeun Hyeon, Sujin Gan, Jueon Kim, Minkyu Kim, Rae Woong Park, and Edward Choi. 2024. Kormedmcqa: Multi-choice question answering benchmark for korean healthcare professional licensing examinations.

Bruce W. Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1281–1309. Association for Computational Linguistics.

LG AI Research. 2024. Komt-bench. https://huggingface.co/datasets/LGAI-EXAONE/KoMT-Bench.

LG AI Research. 2025. EXAONE Deep: Reasoning Enhanced Language Models. *arXiv preprint arXiv:2503.12524*.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024. Datacomp-lm: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious. *CoRR*, abs/2501.00656.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol,

Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas

Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024a. GPT-4o System Card.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. 2024b. Openai o1 system card.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas

Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024c. GPT-4 Technical Report.

Jeonghwan Park. 2024. Logickor.

Sanghee Park and Geewook Kim. 2025. Evaluating multimodal generative AI with Korean educational standards. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 671–688, Albuquerque, New Mexico. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666, Singapore. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi

Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Hyopil Shin, Sangah Lee, Dongjun Jang, Wooseok Song, Jaeyoon Kim, Chaeyoung Oh, Hyemi Jo, Youngchae Ahn, Sihyun Oh, Hyohyeong Chang, Sunkyoung Kim, and Jinsik Lee. 2025. Kobalt: Korean benchmark for advanced linguistic tasks.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I. Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Wei-Yin Ko, Madeline Smith, Antoine Bosselut, Alice Oh, Andre F. T. Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2024. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. KMMLU: Measuring massive multitask language understanding in Korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jae cheol Lee, Je Won Yeom, Jihyu Jung, Jung woo Kim, and Songseong Kim. 2024. HAE-RAE bench: Evaluation of Korean knowledge in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7993–8007, Torino, Italia. ELRA and ICCL.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024a. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *CoRR*, abs/2412.02595.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024b. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *CoRR*, abs/2402.17762.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. 2025. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024a. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024b. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*.

Xiaolin Xing, Zhiwei He, Haoyu Xu, Xing Wang, Rui Wang, and Yu Hong. 2024. Evaluating knowledge-based cross-lingual inconsistency in large language models. *arXiv preprint arXiv:2407.01358*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Mingyu Xu, Xin Men, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. 2024. Base of rope bounds context length. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Greg Yang and Edward J. Hu. 2021. Tensor programs IV: feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML*

2021, 18-24 July 2021, Virtual Event, volume 139 of *Proceedings of Machine Learning Research*, pages 11727–11737. PMLR.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. 2024. Tensor programs VI: feature learning in infinite depth neural networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Haneul Yoo, Cheonbok Park, Sangdoo Yun, Alice Oh, and Hwaran Lee. 2024a. Code-switching curriculum learning for multilingual transfer in llms. *arXiv preprint arXiv:2411.02460*.

Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, Hyunwook Kim, Kyung-Min Kim, Munhyong Kim, Sungju Kim, Donghyun Kwak, Hanock Kwak, Se Jung Kwon, Bado Lee, Dongsoo Lee, Gichang Lee, Jooho Lee, Baeseong Park, Seongjin Shin, Joonsang Yu, Seolki Baek, Sumin Byeon, Eungsup Cho, Dooseok Choe, Jeesung Han, Youngkyun Jin, Hyein Jun, Jaeseung Jung, Chanwoong Kim, Jinhong Kim, Jinuk Kim, Dokyeong Lee, Dongwook Park, Jeong Min Sohn, Sujung Han, Jiae Heo, Sungju Hong, Mina Jeon, Hyunhoon Jung, Jungeun Jung, Wangkyo Jung, Chungjoon Kim, Hyeri Kim, Jonghyun Kim, Min Young Kim, Soeun Lee, Joonhee Park, Jieun Shin, Sojin Yang, Jungsoon Yoon, Hwaran Lee, Sanghwan Bae, Jeehwan Cha, Karl Gylleus, Donghoon Ham, Mihak Hong, Youngki Hong, Yunki Hong, Dahyun Jang, Hyojun Jeon, Yujin Jeon, Yeji Jeong, Myunggeun Ji, Yeguk Jin, Chansong Jo, Shinyoung Joo, Seunghwan Jung, Adrian Jungmyung Kim, Byoung Hoon Kim, Hyomin Kim, Jungwhan Kim, Minkyoung Kim, Minseung Kim, Sungdong Kim, Yonghee Kim, Youngjun Kim, Youngkwan Kim, Donghyeon Ko, Dughyun Lee, Ha Young Lee, Jaehong Lee, Jieun Lee, Jonghyun Lee, Jongjin Lee, Min Young Lee, Yehbin Lee, Taehong Min, Yuri Min, Kiyoon Moon, Hyangnam Oh, Jaesun Park, Kyuyon Park, Younghun Park, Hanbae Seo, Seunghyun Seo, Mihyun Sim, Gyubin Son, Matt Yeo, Kyung Hoon Yeom, Wonjoon Yoo, Myungin You, Doheon Ahn, Homin Ahn, Joohee Ahn, Seongmin Ahn, Chanwoo An, Hyeryun An, Junho An, Sang-Min An, Boram Byun, Eunbin Byun, Jongho Cha, Minji Chang, Seunggyu Chang, Haesong Cho, Youngdo Cho, Dalnim Choi, Daseul Choi, Hyoseok Choi, Minseong Choi, Sangho Choi, Seongjae Choi, Wooyong Choi, Sewhan Chun, Dong Young Go, Chiheon Ham, Danbi Han, Jaemin Han, Moonyoung Hong, Sung Bum Hong, Dong-Hyun Hwang, Seongchan Hwang, Jinbae Im, Hyuk Jin Jang, Jaehyung Jang, Jaeni Jang, Sihyeon Jang, Sungwon Jang, Joonha Jeon, Daun Jeong, Joonhyun Jeong, Kyeongseok Jeong, Mini Jeong, Sol Jin, Hanbyeol Jo, Hanju Jo, Minjung Jo, Chaeyoon Jung, Hyungsik Jung, Jaeuk Jung, Ju Hwan Jung, Kwangsun Jung, Seungjae Jung, Soonwon Ka, Donghan Kang, Soyoung Kang, Taeho Kil, Areum Kim, Beomyoung Kim, Byeongwook Kim, Daehee Kim, Dong-Gyun Kim, Donggook Kim, Donghyun Kim, Euna Kim, Eunchul Kim, Geewook Kim, Gyu Ri Kim, Hanbyul Kim, Heesu Kim, Isaac Kim, Jeonghoon Kim, Jihye Kim, Joonghoon Kim, Minjae Kim, Minsub Kim, Pil Hwan Kim, Sammy Kim, Seokhun Kim, Seonghyeon Kim, Soojin Kim, Soong Kim, Soyoon Kim, Sunyoung Kim, Taeho Kim, Wonho Kim, Yoonsik Kim, You Jin Kim, Yuri Kim, Beomseok Kwon, Ohsung Kwon, Yoo-Hwan Kwon, Anna Lee, Byungwook Lee, Changho Lee, Daun Lee, Dongjae Lee, Ha-Ram Lee, Hodong Lee, Hwiyeong Lee, Hyunmi Lee, Injae Lee, Jaeung Lee, Jeongsang Lee, Jisoo Lee, Jongsoo Lee, Joongjae Lee, Juhan Lee, Jung Hyun Lee, Junghoon Lee, Junwoo Lee, Se Yun Lee, Sujin Lee, Sungjae Lee, Sungwoo Lee, Wonjae Lee, Zoo Hyun Lee, Jong Kun Lim, Kun Lim, Taemin Lim, Nuri Na, Jeongyeon Nam, Kyeong-Min Nam, Yeonseog Noh, Biro Oh, Jung-Sik Oh, Solgil Oh, Yeontaek Oh, Boyoun Park, Cheonbok Park, Dongju Park, Hyeonjin Park, Hyun Tae Park, Hyunjung Park, Jihye Park, Jooseok Park, Junghwan Park, Jungsoo Park, Miru Park, Sang Hee Park, Seunghyun Park, Soyoung Park, Taerim Park, Wonkyeong Park, Hyunjoon Ryu, Jeonghun Ryu, Nahyeon Ryu, Soonshin Seo, Suk Min Seo, Yoonjeong Shim, Kyuyong Shin, Wonkwang Shin, Hyun Sim, Woongseob Sim, Hyejin Soh, Bokyong Son, Hyunjun Son, Seulah Son, Chi-Yun Song, Chiyoung Song, Ka Yeon Song, Minchul Song, Seungmin Song, Jisung Wang, Yonggoo Yeo, Myeong Yeon Yi, Moon Bin Yim, Taehwan Yoo, Youngjoon Yoo, Sungmin Yoon, Young Jin Yoon, Hangyeol Yu, Ui Seon Yu, Xingdong Zuo, Jeongin Bae, Joungeun Bae, Hyunsoo Cho, Seonghyun Cho, Yongjin Cho, Taekyoon Choi, Yera Choi, Jiwan Chung, Zhenghui Han, Byeongho Heo, Euisuk Hong, Taebaek Hwang, Seonyeol Im, Sumin Jegal, Sumin Jeon, Yelim Jeong, Yonghyun Jeong, Can Jiang, Juyong Jiang, Jiho Jin, Ara Jo, Younghyun Jo, Hoyoun Jung, Juyoung Jung, Seunghyeong Kang, Dae Hee Kim, Ginam Kim, Hangyeol Kim, Heeseung Kim, Hyojin Kim, Hyojun Kim, Hyun-Ah Kim, Jeehye Kim, Jin-Hwa Kim, Jiseon Kim, Jonghak Kim, Jung Yoon Kim, Rak Yeong Kim, Seongjin Kim, Seoyoon Kim, Sewon Kim, Sooyoung Kim, Sukyoung Kim, Taeyong Kim, Naeun Ko, Bonseung Koo, Heeyoung Kwak, Haena Kwon, Youngjin Kwon, Boram Lee, Bruce W. Lee,

Dagyeong Lee, Erin Lee, Euijin Lee, Ha Gyeong Lee, Hyojin Lee, Hyunjeong Lee, Jeeyoon Lee, Jeonghyun Lee, Jongheok Lee, Joonhyung Lee, Junhyuk Lee, Mingu Lee, Nayeon Lee, Sangkyu Lee, Se Young Lee, Seulgi Lee, Seung Jin Lee, Suhyeon Lee, Yeonjae Lee, Yesol Lee, Youngbeom Lee, Yujin Lee, Shaodong Li, Tianyu Liu, Seong-Eun Moon, Taehong Moon, Max-Lasse Nihlenramstroem, Wonseok Oh, Yuri Oh, Hongbeen Park, Hyekyung Park, Jaeho Park, Nohil Park, Sangjin Park, Jiwon Ryu, Miru Ryu, Simo Ryu, Ahreum Seo, Hee Seo, Kangdeok Seo, Jamin Shin, Seungyoun Shin, Heetae Sin, Jiangping Wang, Lei Wang, Ning Xiang, Longxiang Xiao, Jing Xu, Seonyeong Yi, Haanju Yoo, Haneul Yoo, Hwanhee Yoo, Liang Yu, Youngjae Yu, Weijie Yuan, Bo Zeng, Qian Zhou, Kyunghyun Cho, Jung-Woo Ha, Joonsuk Park, Jihyun Hwang, Hyoung Jo Kwon, Soonyong Kwon, Jungyeon Lee, Seungho Lee, Seonghyeon Lim, Hyunkyung Noh, Seungho Choi, Sang-Woo Lee, Jung Hwa Lim, and Nako Sung. 2024b. Hyperclova x technical report.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825.

Yonghao Zhuang, Lanxiang Hu, Longfei Yun, Souvik Kundu, Zhengzhong Liu, Eric P. Xing, and Hao Zhang. 2025. Scaling long context training data by long-distance referrals. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

A Contributors

Within each role, names are listed in alphabetical order by last name, followed by the first name.

Core Contributors

Sanghwan Bae
Minseong Choi
Hyunsoo Ha
Chiheon Ham
Donghoon Ham
Jaemin Han
Jiwoo Hong†
Youngki Hong
Jinbae Im
Sookyo In
Yeguk Jin
Chansong Jo
Hwiyeol Jo
Shinyoung Joo
Jingu Kang
Donghyeon Ko
Taeho Kil
Byeongwook Kim
Daehee Kim
Donghyun Kim
Geewook Kim
Hanbyul Kim
Hyunwoo Kim
Jeonghoon Kim
Jungwhan Kim
Minkyoung Kim
Munhyong Kim
Seonghyun Kim
Sungdong Kim†
Sungju Kim
Yoonsik Kim
You Jin Kim
Donghyun Kwak
Beomseok Kwon
Bado Lee
Byungwook Lee
Gichang Lee
Hodong Lee
Injae Lee
Jaehong Lee
Jeong Hyun Lee†
Jieun Lee
Joosung Lee
Min Young Lee
Noah Lee
Sang-Woo Lee†
Yehbin Lee
Yujeong Lee
Taehong Min
Kiyoon Moon
JeongYeon Nam
Yeontaek Oh
Cheonbok Park
Joonsuk Park
Kyuyon Park
Sanghee Park
Ahreum Seo
Seunghyun Seo
Suk Min Seo
Seongjin Shin
Ka Yeon Song
Nako Sung
Moonbin Yim
Kang Min Yoo
Taehwan Yoo
MyungIn You
Hangyeol Yu

Contributors

Sang Min An
Jeongin Bae
Chongho Cha
Eungsup Cho
Haesong Cho
Saerim Cho
Hyungwook Choi
Jaepil Choi †
Sanghyuk Choi
Jaehyeok Doo †
Sungbum Hong
Seongchan Hwang
Donghoon Jang
Genie Jang
Junseo Jang
Heewon Jeon
Mina Jeon
Kyeongseok Jeong
Yelim Jeong
Myunggeun Ji
Youngkyun Jin
Ara Jo
Hyunhoon Jung
Kwangsun Jung
Seunghwan Jung
Dain Kim †
Dong Gyun Kim
Eunchul Kim
Ginam Kim
Hyomin Kim
Hyunwook Kim
Jihye Kim
Jiseob Kim
Jonghak Kim
Joonghoon Kim †
Minseung Kim
Minyoung Kim
Singon Kim

Soyoon Kim
Taeyong Kim
Yonghee Kim
Youngjun Kim
Ohsung Kwon
Yoo Hwan Kwon
Youngjin Kwon
Dagyeong Lee
Dughyun Lee
Gayoung Lee
Ha Ram Lee
Hagyeong Lee
Jeonghyun Lee
Jonghyun Lee
Jongjin Lee
Joonhyung Lee
Junghoon Lee
Seulgi Lee
Soeun Lee
Sujin Lee
Sungwoo Lee
Yesol Lee
Youngbeom Lee
Taemin Lim
Kyeong Min Nam
Biro Oh
Solgil Oh
Gunho Park
Wonkyeong Park
Jieun Shin
Wonkwang Shin
Chiyun Song
Hae Jin Song
Minchul Song
Sukwon Yeo
Hwanhee Yoo
Wonjoon You
Uiseon Yu

[†] Work done while at NAVER Cloud.

B Performance on Math&Coding Benchmarks

Category	Benchmarks	HCX THINK (-)	Qwen3 (32B)	Qwen3 (14B)	QwQ (32B)	EXAONE Deep (32B)
Math	**GSM8K**	95.5	95.9	95.9	**96.2**	95.5
	GSM8K-ko	92.1	**93.4**	92.6	92.7	91.8
	MATH500	95.2	97.2	96.8	**98.0**	96.0
	MATH100-ko	90.5	92.9	92.9	92.9	**93.9**
Coding	**HumanEval**	95.7	**96.9**	95.7	90.8	95.1
	MBPP	89.9	78.6	**90.8**	90.1	87.5

Table 6: Performance comparison of language models on math and coding benchmarks. Differences among the baseline reasoning models are marginal, as these benchmarks are approaching saturation.

C KoBigBench Qualitative Examples

HyperCLOVA X THINK (Translated Model Input&Output – English)

<|im_start|>tool_list
<|im_end|>
<|im_start|>system
The following are multiple choice questions with explanations and/or answers about social-ecofin201.<|im_end|>
<|im_start|>user
Question: What is the most reasonable measure the government can take to prevent the "transaction cliff" shock in the real estate market?
Read the question again: What is the most reasonable measure the government can take to prevent the "transaction cliff" shock in the real estate market?
Options:
(A). Increase real estate transaction tax rate
(B). Increase property tax rate
(C). Strengthen safety inspections for reconstruction
(D). Lift designation of regulated areas<|im_end|>
<|im_start|>assistant/think
This question asks for the most reasonable measure the government can take to address the "transaction cliff" problem in the real estate market today. We have four options from A to D. I need to examine each option to see which one would help prevent a transaction cliff.

First, I need to understand exactly what a "transaction cliff" is. It seems to refer to a phenomenon where real estate transaction volume rapidly decreases. This term is typically used when the real estate market is in a slump or when regulations are tightened, leading to a contraction in transactions. Therefore, to prevent this, the government would likely need to implement policies that stimulate transactions.

Let's analyze each option one by one.

(A) Increase real estate transaction tax rate: Transaction taxes refer to taxes like acquisition tax or registration tax paid when buying or selling real estate. Increasing the transaction tax rate would increase transaction costs, which would likely further reduce transactions. This would worsen the transaction cliff rather than prevent it. So, A does not seem like an

appropriate measure.

(B) Increase property tax rate: Property tax refers to taxes paid while owning a home, such as property tax or comprehensive real estate tax. Increasing property taxes might lead some homeowners to put their properties on the market due to the tax burden. However, conversely, existing owners might also choose to hold onto their properties. It's also somewhat ambiguous whether an increase in property taxes would directly stimulate transactions. There's also an argument that increasing property taxes could lead to a decrease in transactions itself. So, B also seems inappropriate for preventing a transaction cliff.

(C) Strengthen safety inspections for reconstruction: Strengthening safety inspections for reconstruction would reduce the supply through redevelopment. However, whether this is relevant depends on whether the transaction cliff is due to a supply shortage or other factors. If the lack of transactions is due to insufficient new supply through reconstruction, then relaxation, not strengthening, might be needed. But the question asks for a measure to prevent a transaction cliff, and strengthening safety inspections for reconstruction could further reduce supply, potentially negatively impacting the market. Therefore, C also seems inappropriate.

(D) Lift designation of regulated areas: Regulated areas refer to regions where subscription, loan, and resale restrictions apply. Lifting the designation of regulated areas would ease transaction regulations in those areas, which could stimulate transactions, right? For example, if loan regulations are eased or resale restrictions are removed, it would become easier to conduct transactions, potentially increasing transaction volume. Therefore, lifting the designation of regulated areas could help prevent a transaction cliff.

However, even with this thought, I need to consider more specifically how lifting the designation of regulated areas actually impacts transaction activation. If regulated areas are lifted, they would be removed from speculative overheated zones or adjustment target areas, leading to relaxed subscription conditions, increased loan limits, and even allowing multiple homeowners to get loans, thereby easing transaction-related regulations. This could increase buying demand and lead to more active transactions. On the other hand, there might be concerns that lifting regulated areas could induce speculation and cause market overheating. However, the question specifically asks for a measure to prevent a transaction cliff, i.e., a rapid decrease in transactions, so regulatory easing seems appropriate.

Comparing each option, A is inappropriate because it increases transaction costs and reduces transactions. B might lead to property sales due to increased holding tax but is less directly related to transaction activation. C could lead to reduced supply and further decrease transactions. D could promote transactions by easing regulations, so D seems to be the most reasonable measure.

However, one lingering question is whether lifting the designation of regulated areas can be a direct solution to the transaction cliff. For example, if the current real estate market is in a slump and transactions have decreased, easing regulations could stimulate demand and increase transactions. Conversely, if the market was overheated and regulations were tightened, easing them could lead to re-overheating, but the question is about preventing a transaction cliff, a decrease in transactions, so regulatory easing seems correct.

Also, another possibility is that if the transaction cliff is due to external factors like rising interest rates or an economic downturn, the government's regulatory easing alone might have limited effect. However, the question asks to choose from government measures, so it's about selecting the most appropriate policy tool.

In summary, to increase transactions, lowering transaction costs or easing regulations would be helpful. Increasing the transaction tax rate (A) increases costs and is therefore opposite. Increasing the property tax rate (B) increases the burden of ownership, which

might incentivize selling, but it's less directly linked to transaction activation. Strengthening safety inspections for reconstruction (C) could lead to a decrease in supply and thus negatively impact transaction activation. Lifting the designation of regulated areas (D) eases regulations and can promote transactions, so it seems to be the most appropriate measure.

However, I need to re-evaluate whether the assumption that lifting the designation of regulated areas helps activate transactions is correct. Regulated areas are designated as speculative overheated or adjustment target areas. Lifting these designations can ease loan regulations, resale restrictions, etc., making transactions easier. For example, if the resale restriction period is shortened, short-term trading becomes possible, which can increase transaction volume. If loan regulations are eased, real demanders can more easily access housing purchases, which can stimulate transactions. Therefore, lifting the designation of regulated areas can help alleviate the transaction cliff.

On the other hand, if the current transaction cliff is due to reduced liquidity from interest rate hikes or decreased demand due to an economic downturn, even regulatory easing might not be enough to increase transactions. However, the question asks for a government measure to prevent the transaction cliff in the real estate market, so among the given options, D seems the most appropriate.

Let's double-check the other options.

(A) Increase real estate transaction tax rate: This would definitely increase transaction costs and reduce transactions, thus deepening the transaction cliff rather than preventing it. So, A is incorrect.

(B) Increase property tax rate: While some argue that increasing property taxes might make homeowners feel the tax burden and put properties on the market, conversely, they might also hold onto their properties. Also, increasing property taxes primarily affects multi-homeowners or owners of high-value properties. While their putting properties on the market could increase supply and activate transactions, this can vary depending on the situation. However, generally, an increase in property taxes is more focused on price stabilization through increased supply rather than transaction activation. Therefore, B might not be an appropriate direct measure to solve the transaction cliff itself.

(C) Strengthen safety inspections for reconstruction: Expanding supply through reconstruction can contribute to market stability by increasing housing supply. However, strengthening safety inspections makes reconstruction more difficult, thus reducing supply. Reduced supply can lead to upward price pressure but is a separate issue from transaction activation. If the lack of transactions is due to a supply shortage, then supply needs to be increased, but if the current transaction cliff is due to other factors, this measure would be ineffective. Therefore, C seems inappropriate for activating transactions.

(D) Lift designation of regulated areas: As I considered earlier, easing regulations can make transactions easier, potentially increasing transaction volume, so D seems the most appropriate.

Furthermore, recalling the actual impact of lifting regulated area designations on transaction activation in Korean real estate policy, for example, the government lifted some regulated areas in late 2022 or early 2023. At that time, the real estate market was cooling, and the purpose was to stimulate transactions by easing regulations. Therefore, in actual cases, lifting regulated area designations has been used as a policy to alleviate transaction cliffs, which increases the likelihood that D is the correct answer in this problem.

In conclusion, analyzing each option, (D) Lifting designation of regulated areas seems to be the most reasonable measure. <|im_end|> <|im_start|>assistant
Answer: D<|im_end|>

HyperCLOVA X THINK (Model Input&Output – Korean)

<|im_start|>tool_list
<|im_end|> <|im_start|>system
The following are multiple choice questions with explanations and/or answers about social-ecofin201.<|im_end|>
<|im_start|>user
Question: 부동산 시장의 '거래 절벽' 충격을 막기 위해 정부가 취할 수 있는 조치로 가장 합리적인 것은?
Read the question again: 부동산 시장의 '거래 절벽' 충격을 막기 위해 정부가 취할 수 있는 조치로 가장 합리적인 것은?
Options:
(A). 거래세율 인상
(B). 보유세율 인상
(C). 재건축 안전진단 강화
(D). 규제지역 해제 <|im_end|>
<|im_start|>assistant/think
오늘 부동산 시장의 '거래 절벽' 문제를 해결하기 위해 정부가 취할 수 있는 조치 중에서 가장 합리적인 것을 고르는 문제네요. 문제는 선택지 A부터 D까지 네 가지 옵션이 주어졌어요. 각 옵션을 하나씩 살펴보면서 어떤 것이 거래 절벽을 막는 데 도움이 될지 생각해봐야겠어요.

먼저, '거래 절벽'이 무엇인지부터 정확히 이해해야 할 것 같아요. 거래 절벽이란 부동산 거래량이 급격히 감소하는 현상을 말하는 것 같은데, 보통 부동산 시장이 침체기에 빠지거나 규제가 강화되어 거래가 위축될 때 사용되는 용어인 것 같아요. 그렇다면 정부가 이를 막기 위해서는 거래를 활성화시킬 수 있는 정책을 펴야 하지 않을까 싶네요.

각 선택지를 하나씩 분석해 볼게요.

(A) 거래세율 인상: 거래세란 부동산을 사고 팔 때 내는 취득세나 등록세 같은 것을 말하는 것 같아요. 거래세율을 인상하면 거래 비용이 늘어나게 되므로, 거래를 더 줄이는 효과가 있을 것 같은데. 그러면 거래 절벽을 막기보다는 오히려 악화시키지 않을까? 따라서 A는 적절한 조치가 아닐 것 같아요.

(B) 보유세율 인상: 보유세는 주택을 소유하고 있는 동안 내는 세금, 예를 들어 재산세나 종합부동산세 같은 걸 말하죠. 보유세를 올리면 집을 가진 사람들이 세금 부담을 느껴 매물을 내놓을 수도 있겠지만, 반대로 생각하면 기존 소유자들이 매물을 거둬들이게 될 수도 있을 것 같아요. 하지만 보유세 인상이 거래 활성화에 직접적인 영향을 미칠지는 좀 애매한 것 같아요. 특히 보유세를 올리면 거래 자체가 줄어들 수 있다는 의견도 있을 것 같고. 그래서 B도 거래 절벽을 막는 데는 부적절할 것 같네요.

(C) 재건축 안전진단 강화: 재건축 안전진단을 강화하면 재건축을 통한 공급이 줄어들게 되겠죠. 그런데 거래 절벽이 공급 부족으로 인한 것인지, 아니면 다른 요인 때문인지에 따라 다를 것 같아요. 만약 재건축을 통한 신규 공급이 부족해서 거래가 안 된다면 강화가 아니라 완화가 필요할 수도 있지 않을까요? 하지만 문제에서는 거래 절벽을 막기 위한 조치라고 했으니, 재건축 안전진단 강화는 공급을 더 줄여서 오히려 시장에 부정적인 영향을 줄 수도 있을 것 같아요. 따라서 C도 적절하지 않을 것 같네요.

(D) 규제지역 해제: 규제지역이란 청약, 대출, 전매 제한 등이 적용되는 지역을 말하는 것 같아요. 규제지역을 해제하면 해당 지역에서의 거래 규제가 완화되어 거래가 활성화될 수 있지 않을까? 예를 들어, 대출 규제가 완화되거나 전매 제한이 없어지면 거래를 하기 쉬워져서 거래량이 늘어날 수 있을 것 같아요. 따라서 규제지역 해제는 거래 절벽을 막는 데 도움이 될 수 있을 것 같네요.

하지만 이렇게 생각해보지만, 실제로 규제지역 해제가 거래 활성화에 어떤 영향을 미치는지 좀 더 구체적으로 알아볼 필요가 있을 것 같아요. 규제지역이 해제되면 투기과열지구, 조정대상지역 등에서 벗어나게 되어서 청약 조건이 완화되고, 대출 한도가 늘어나거나, 다주택자도 대출을 받을 수 있게 되는 등 거래 관련 규제가 완화되겠죠. 이렇게 되면

매수 수요가 늘어나 거래가 활발해질 수 있을 것 같아요. 반면에 규제지역 해제가 오히려 투기를 유발하여 시장 과열을 초래할 수도 있다는 우려도 있을 수 있지만, 문제에서는 거래 절벽, 즉 거래가 급격히 줄어드는 현상을 막기 위한 조치이므로 규제 완화가 도움이 될 것 같아요.

그래서 각 선택지를 비교해보면, A는 거래 비용을 늘려 거래를 줄이므로 부적절하고, B는 보유세 인상으로 매물 출회가 있을 수도 있지만 직접적인 거래 활성화보다는 다른 효과를 가져올 수 있으며, C는 공급 감소로 거래를 더 줄일 수 있고, D는 규제 완화로 거래를 촉진할 수 있으므로 가장 합리적인 것은 D가 아닐까 생각해요.

하지만 여기서 한 가지 의문이 드는 점이 있어요. 규제지역 해제가 거래 절벽을 해결하는 직접적인 방법이 될 수 있을까요? 예를 들어, 현재 부동산 시장이 침체기라서 거래가 줄어든 상황이라면, 규제를 풀어주는 것이 수요를 자극하여 거래를 늘릴 수 있을 거예요. 반면에 시장이 과열되어서 규제를 강화한 상황이라면, 규제를 완화하면 다시 과열될 수 있지만, 문제에서는 거래 절벽, 즉 거래 감소 상황을 막기 위한 것이므로 규제 완화가 맞을 것 같아요.

또한 다른 가능성으로는, 거래 절벽이 금리 상승이나 경기 침체 등 외부 요인에 의한 것이라면 정부의 규제 완화만으로는 효과가 제한적일 수 있지만, 문제에서는 정부의 정책 중에서 선택하라고 했으니 정책적 수단 중에서 가장 적절한 것을 고르는 문제일 거예요.

정리하자면, 거래를 늘리기 위해서는 거래 비용을 낮추거나 규제를 완화하는 것이 도움이 될 텐데, 거래세율 인상(A)은 비용을 높이므로 반대이고, 보유세 인상(B)은 보유 부담을 높여 매도 유인은 있지만 거래 활성화와는 직접 연관이 적을 것 같아요. 재건축 안전진단 강화(C)는 공급 감소로 작용할 수 있으니 역시 거래 활성화에 부정적일 것 같고, 규제지역 해제(D)는 규제를 완화하여 거래를 촉진할 수 있으므로 가장 적절한 조치로 보여요.

하지만 여기서 규제지역 해제가 거래 활성화에 도움이 된다는 가정이 맞는지 다시 한 번 생각해봐야 할 것 같아요. 규제지역으로 지정된 지역은 투기과열이나 조정대상지역 등인데, 이 지역들을 해제하면 대출 규제, 전매 제한 등이 완화되어 거래가 용이해질 수 있습니다. 예를 들어, 전매 제한 기간이 줄어들면 단기 매매가 가능해져서 거래량이 늘어날 수 있고, 대출 규제가 완화되면 실수요자들이 주택 구입에 더 쉽게 접근할 수 있게 되어 거래가 활성화될 수 있겠죠. 따라서 규제지역 해제는 거래 절벽을 완화하는 데 도움이 될 수 있을 것 같아요.

반면에, 만약 현재 거래 절벽이 금리 인상으로 인한 유동성 축소나 경기 침체에 따른 수요 감소 때문이라면, 규제 완화로도 거래를 늘리기 어려울 수 있지만, 문제에서는 부동산 시장의 거래 절벽을 막기 위한 정부의 조치 중에서 선택하라고 했으므로, 주어진 선택지 중에서는 D가 가장 적절하다고 봐야 할 것 같아요.

그럼 다른 선택지들을 다시 한 번 체크해보죠.

(A) 거래세율 인상: 이건 확실히 거래 비용을 증가시켜 거래를 줄이므로, 거래 절벽을 막기보다는 심화시킬 것 같아요. 따라서 A는 틀린 답.

(B) 보유세율 인상: 보유세를 올리면 집주인들이 세금 부담을 느껴 매물을 내놓을 수 있다는 의견도 있지만, 반대로 매물을 내놓지 않고 버티기를 할 수도 있어요. 또한, 보유세 인상은 주로 다주택자나 고가 주택 소유자에게 타격을 줄 텐데, 이들이 매물을 내놓으면 공급이 늘어나 거래가 활성화될 수도 있겠지만, 이는 상황에 따라 다를 수 있어요. 하지만 일반적으로 보유세 인상은 거래 활성화보다는 공급 증가를 통한 가격 안정화에 더 초점이 맞춰져 있을 것 같아요. 따라서 거래 절벽 자체를 해결하기 위한 직접적인 조치로는 B가 적절하지 않을 수 있어요.

(C) 재건축 안전진단 강화: 재건축을 통한 공급 확대는 주택 공급을 늘려서 시장 안정에 기여할 수 있지만, 안전진단을 강화하면 재건축이 어려워져서 공급이 줄어들게 되죠. 공급 감소는 가격 상승 압력으로 작용할 수 있지만, 거래 활성화와는 별개 문제일 수

있어요. 만약 공급이 부족해서 거래가 안 된다면 공급을 늘려야 하지만, 현재 거래 절벽은 공급 부족보다는 다른 요인 때문이라면 이 조치는 효과가 없을 거예요. 따라서 C는 거래 활성화를 위한 조치로는 부적절해 보여요.

(D) 규제지역 해제: 앞서 생각한 것처럼 규제를 완화하면 거래가 용이해져서 거래량이 늘어날 수 있으므로 D가 가장 적절하다고 생각됩니다.

하지만 실제로 한국의 부동산 정책에서 규제지역 해제가 거래 활성화에 미친 영향은 어땠는지 기억을 더듬어 보면, 예를 들어 2022년 말이나 2023년 초에 정부가 일부 규제지역을 해제한 적이 있었어요. 그 당시에는 부동산 시장이 냉각되어 있었기 때문에 규제를 완화해서 거래를 촉진하려는 목적이었을 거예요. 따라서 실제 사례에서도 규제지역 해제가 거래 절벽 완화에 도움이 되는 정책으로 사용된 바 있으니, 이 문제에서도 D가 정답일 가능성이 높아 보여요.

결론적으로, 각 옵션을 분석해보면 (D) 규제지역 해제가 가장 합리적인 조치로 보입니다.<|im_end|> <|im_start|>assistant Answer: D<|im_end|>

HyperCLOVA X THINK (Translated Model Input&Output – English)

<|im_start|>tool_list
<|im_end|> <|im_start|>system
The following are multiple choice questions with explanations and/or answers about commonsense-context30.<|im_end|>
<|im_start|>user
Question: Context: Everyone eventually came back to me and made me feel loved. It was truly wonderful. It's amazing and touching that I have such wonderful friends by my side. I think it felt even more special because I hadn't seen them in such a long time. It's strangely surprising to have returned like this, but meeting people again is really a great thing. Especially Donghwan.
Question: If I hadn't left, how would I have felt?
Read the question again: Context: Everyone eventually came back to me and made me feel loved. It was truly wonderful. It's amazing and touching that I have such wonderful friends by my side. I think it felt even more special because I hadn't seen them in such a long time. It's strangely surprising to have returned like this, but meeting people again is really a great thing. Especially Donghwan.
Question: If I hadn't left, how would I have felt?
Options:
(A). Donghwan would have hated me, and nothing would have happened.
(B). I wouldn't have missed these people this much.
(C). It's still joyful and touching to have friends who hurt me.
(D). It would have felt strange not to have left home at all.<|im_end|>
<|im_start|>assistant/think
Today, the user provided context and a question, requiring me to select the most appropriate answer. I'll first carefully examine the context provided. The speaker mentions feeling loved and touched because friends returned to their side, emphasizing how special it felt due to the long absence, especially highlighting Donghwan. The speaker mentions that returning felt strangely surprising but meeting people again was truly great.

The question asks, "If I hadn't left, how would I have felt?" Currently, the speaker is moved because of the reunion after leaving and returning. If they had never left, the intense feelings related to the reunion might not have existed.

Analyzing each option individually:

(A). Donghwan would have hated me, and nothing would have happened.
This doesn't align with the context. There is no indication in the context that Donghwan or

others would have felt negatively if the speaker hadn't left.

(B). I wouldn't have missed these people this much.
This is logical. The speaker emphasizes that the long absence made the reunion more special and touching. Without leaving, the feeling of longing would naturally be less intense.

(C). It's still joyful and touching to have friends who hurt me.
This is incorrect. The context never indicates that the friends were hurtful. Rather, it focuses on positive and warm feelings upon reunion.

(D). It would have felt strange not to have left home at all.
The context describes the return as "strangely surprising," suggesting that leaving and returning was unusual. If the speaker had never left, it might have felt normal rather than strange to remain home. Therefore, option D is somewhat ambiguous but doesn't strongly align with the context.

Given the context and the reasoning above, option (B) stands out as the most appropriate choice. The emotional impact of missing friends after a long absence is clearly highlighted, and thus, had the speaker never left, these strong feelings of longing would not have existed.

Hence, the correct answer is option (B).<|im_end|>
<|im_start|>assistant
Answer: B<|im_end|>

HyperCLOVA X THINK (Model Input&Output – Korean)

<|im_start|>tool_list
<|im_end|><|im_start|>system
The following are multiple choice questions with explanations and/or answers about commonsense-context30.<|im_end|>
<|im_start|>user
Question: 맥락: 모두가 결국 저한테 돌아와줬고 제가 사랑받고 있다고 느끼게 했어요. 진짜 최고였네요. 진짜 이런 멋진 친구들이 제 곁에 있다는게 너무 좋고 감동적이에요. 특히 오랫동안 못봐서 더 그랬던거 같아요. 이렇게 돌아오게 된게 이상할 정도지만 사람들이랑 다시 만난다는 것은 진짜 좋은 일이에요. 특히 동환이요.
질문: 만약 내가 떠나지 않았더라면 내가 어떻게 느꼈을까?
Read the question again: 맥락: 모두가 결국 저한테 돌아와줬고 제가 사랑받고 있다고 느끼게 했어요. 진짜 최고였네요. 진짜 이런 멋진 친구들이 제 곁에 있다는게 너무 좋고 감동적이에요. 특히 오랫동안 못봐서 더 그랬던거 같아요. 이렇게 돌아오게 된게 이상할 정도지만 사람들이랑 다시 만난다는 것은 진짜 좋은 일이에요. 특히 동환이요.
질문: 만약 내가 떠나지 않았더라면 내가 어떻게 느꼈을까?
Options:
(A). 동환이가 나를 미워했을거라 아무 일도 없었을 것이다.
(B). 이만큼 이 사람들을 그리워하진 않았을 것이다.
(C). 상처 주는 친구들이 있다는 것이 여전히 기쁘고 감동적이다.
(D). 집을 전혀 떠나지 않았다는 것이 이상하게 느껴졌을 것이다.<|im_end|>
<|im_start|>assistant/think
오늘 사용자가 맥락과 질문을 주고, 그에 맞는 답을 선택하는 문제를 내주었어. 먼저 맥락을 자세히 읽어봐야겠지. 맥락을 보면, 주인공은 친구들이 돌아와주고 사랑받는다고 느껴서 감동적이고 좋았다고 말하고 있어. 특히 오랫동안 못 봤던 친구들, 특히 동환이를 다시 만나게 되어서 더 그랬다고 해. 그리고 "이렇게 돌아오게 된게 이상할 정도지만" 이라고 했어. 이게 중요한 부분일 것 같아.

질문은 "만약 내가 떠나지 않았더라면 내가 어떻게 느꼈을까?"야. 현재 상황은 주인공이 떠났다가 다시 돌아왔고, 그로 인해 친구들이 돌아와줘서 감동받은 상황이야. 그런데 만약 떠나지 않았다면 어떻게 느꼈을까를 묻는 거지.

선택지를 하나씩 분석해봐야겠어.

(A) 동환이가 나를 미워했을거라 아무 일도 없었을 것이다.
이건 맥락과 좀 안 맞는 것 같아. 현재 상황은 친구들이 돌아와줘서 좋은 상황인데, 떠나지 않았다면 동환이가 미워했을 거라니? 맥락에서는 떠나서 다시 돌아왔을 때 친구들이 돌아와줬으니, 떠나지 않았다면 친구들이 떠나지 않았을까? 그런데 이건 좀 헷갈리네. 아니면 주인공이 떠났기 때문에 친구들이 돌아온 것일까? 맥락에서 "모두가 결국 저한테 돌아와줬고"라고 했으니까, 주인공이 떠난 후 친구들이 돌아온 걸 수도 있겠지. 그런데 만약 떠나지 않았다면 친구들이 돌아올 필요가 없었을 거고, 그래서 그 감동을 느끼지 못했을 수도 있겠다. 그런데 (A)는 동환이가 미워했을 거라고 하니 맥락과 연결이 잘 안 돼 보여.

(B) 이만큼 이 사람들을 그리워하진 않았을 것이다. 이건 가능할 것 같아. 주인공이 떠났기 때문에 친구들을 오랫동안 못 봤고, 그래서 다시 만났을 때 더 감동적이었던 거잖아. 만약 떠나지 않았다면 오랫동안 못 보지 않았을 테니 그렇게까지 그리워하지 않았을 거라는 말이 되겠지. 맥락에서 "특히 오랫동안 못봐서 더 그랬던거 같아요"라고 했으니, 떠나지 않았다면 그 부분이 없었을 테니까. 이 선택지는 타당해 보여.

(C) 상처 주는 친구들이 있다는 것이 여전히 기쁘고 감동적이다.
이건 맥락과 맞지 않아. 현재 상황은 친구들이 돌아와서 감동적이지만, 상처 주는 친구들이라면? 맥락에서는 친구들이 돌아와서 좋은 상황이니까, 상처 주는 친구들이 있다는 게 아니라 오히려 돌아와서 좋은 상황이잖아. 따라서 (C)는 부적절한 것 같아.

(D) 집을 전혀 떠나지 않았다는 것이 이상하게 느껴졌을 것이다.
이건 좀 애매해. 현재 상황에서 "이렇게 돌아오게 된게 이상할 정도지만 사람들이랑 다시 만난다는 것은 진짜 좋은 일이에요"라고 했으니까, 돌아오는 게 이상했지만 좋은 일이라고 했어. 만약 집을 떠나지 않았다면, 돌아오는 상황 자체가 없으니까, "집을 전혀 떠나지 않았다는 것이 이상하게 느껴졌을 것이다"라고 하면, 현재 상황에서 집을 떠난 것이 이상했지만 좋은 거라면, 떠나지 않았을 때는 그 이상함이 없으니 이상하게 느껴지지 않을까? 그런데 이건 좀 추측이 필요해. 맥락에서 "이렇게 돌아오게 된게 이상할 정도"라고 했으니까, 떠나지 않았다면 돌아올 필요가 없으니 그 이상함은 없었을 거고, 따라서 집을 떠나지 않았다는 것이 이상하게 느껴지지 않을 것 같은데. 아니면 원래 집을 떠나는 게 일반적인 상황이라면, 떠나지 않았을 때 오히려 이상하게 느껴질 수도 있을까? 이건 좀 애매한 것 같아.

다시 질문으로 돌아가서, "만약 내가 떠나지 않았더라면 내가 어떻게 느꼈을까?"라고 했을 때, 현재 상황은 떠났다가 돌아와서 친구들을 다시 만나 감동받았어. 따라서 떠나지 않았다면 친구들과 떨어져 있지 않았을 테니, 다시 만나는 감동은 없었을 거야. 특히 오랫동안 못 봤기 때문에 더 감동적이었던 부분이 있으므로, 떠나지 않았다면 그만큼 그리워하지 않았을 것(B)이라는 말이 맞을 것 같아.

(A)는 동환이가 미워했을 거라니 맥락과 안 맞고, (C)는 상처 주는 친구들이라는 게 말이 안 되고, (D)는 집을 떠나지 않은 것이 이상하게 느껴진다는 건데, 현재 상황에서 집을 떠난 것이 이상했지만 좋은 거라면, 떠나지 않았을 때는 그 이상함이 없으니 이상하지 않을 것 같은데. 따라서 (D)는 애매하고, (B)가 가장 자연스러워 보여.

하지만 맥락에서 "이렇게 돌아오게 된게 이상할 정도지만 사람들이랑 다시 만난다는 것은 진짜 좋은 일이에요"라고 했으니, 돌아오는 게 예상 밖의 일이었지만 좋은 일이었다는 거지. 만약 떠나지 않았다면 돌아올 필요가 없으니, 돌아올 일이 없어서 그 좋은 일을 경험하지 못했을 거야. 따라서 "이만큼 이 사람들을 그리워하진 않았을 것이다"라는 (B)가 맞을 것 같아. 왜냐하면 떠나서 오랫동안 못 봤기 때문에 그리워했고, 다시 만나서 감동적이었으니까, 떠나지 않았다면 그렇게까지 그리워하지 않았을 테니까.

그래서 정답은 (B)인 것 같아.<|im_end|>
<|im_start|>assistant
Answer: B<|im_end|>

D Qualitative Examples in Vision Adaptation Ablation

We present below a representative example from the KCSAT STEM vision benchmark. **All problems are rendered and presented to the model strictly as images**, without any accompanying text input. This setup requires the model to make use of its full vision capability–not only to recognize and interpret textual content within the images, but also to read and reason about mathematical expressions, tables, and diagrams where present. For all examples, the user query is standardized as: **"이미지를 보고 문제를 풀어주세요."** ("Please solve the problem by looking at the image.")

D.1 KCSAT Biology Basic Example

We present our model's output on a sample KCSAT Biology Basic problem. Figure 7 shows the original problem image, which is directly provided to the model for analysis and reasoning. The model is required to interpret, reason through, and solve the problem solely based on the image. For reference, we include the model's original answer, as well as an English translation of the problem for readers. A literal English translation of the model's output is also provided for comparison.

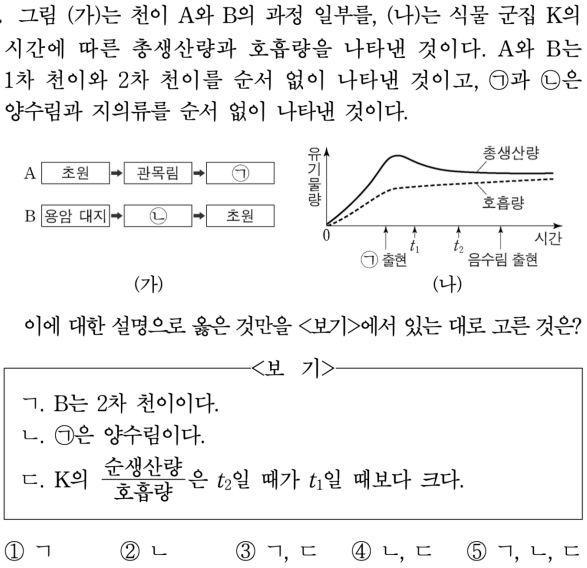


Figure 7: Original KCSAT Sample (in Korean).

Sample Question 8 (Translated)
The diagrams below represent parts of processes A and B in succession, and a graph depicting the changes in total productivity and respiration amounts of plant community K over time. A and B refer to primary and secondary successions in no particular order, and labels ㄱ and ㄴ refer to hydrosere and xerosere in no particular order.

<Process Diagram> <Graph>

Which of the following statements are correct based on the above?

(Options)

(a) B is a secondary succession.
(b) ㄱ is a hydrosere.
(c) For K, the $\frac{\text{total productivity}}{\text{respiration}}$ is greater at t_2 than at t_1.

Select the correct combination(s):

(1) a (2) b (3) a, c (4) b, c (5) a, b, c

HyperCLOVA X THINK (Translated Model Output – English)

<|im_start|>assistant/think
Okay, let's solve this problem. First, we need to look carefully at the figure given in the problem.
First, we need to examine figures (a) and (b) of question 8. According to the problem, A and B are one of primary succession and secondary succession, and ㉠ and ㉡ represent deciduous forest and lichen respectively, in no particular order. Also, it shows a graph of the changes in total productivity and respiration amount of plant community K over time.
Let's look at (a) first. A is said to go from grassland to shrub forest to ㉠, and B is said to start from lava field, go through ㉡, and reach grassland. Here, we need to figure out which succession process A and B each represent. Also, we need to identify what ㉠ and ㉡ are.
If we think about the succession process, primary succession starts in lifeless areas, and secondary succession is the process of recovery after disturbance in an area where plants already existed. For example, starting on bare land created by disturbances like forest fires or logging is secondary succession. On the other hand, primary succession starts in places like rocks or lava fields, where there isn't even soil.
In the problem, since A starts from grassland (a place where plants already exist), this would be the secondary succession process. That's because grassland is an environment with soil and some living organisms. In contrast, B starts from a lava field, which is formed by volcanic activity and likely has no soil. Thus, B would correspond to primary succession. Therefore, the conclusion is that A is secondary succession and B is primary succession... but wait, in the problem it states that A and B represent "primary and secondary in no order," so I might have thought in reverse, so confirmation is necessary.
Looking again, A: grassland → shrub forest → ㉠
B: lava field → ㉡ → grassland
A already begins with grassland, so it's a secondary succession process, and B starts from lava field, so it is a primary succession process. So A is secondary, B is primary. Then in the choices, statement ㄱ says "B is secondary succession," so that's a wrong statement. So ㄱ is incorrect.
Next, let's look at what ㉠ and ㉡ are. A process is secondary succession, and generally, the final stage of secondary succession could be a climax, such as a shade-tolerant forest, but depending on the case, it could be something else. However, usually, secondary succession develops into a more complex forest structure over time. But here, A passes through shrub forest and goes to ㉠, so if ㉠ is a deciduous forest, the next stage would be a shade-tolerant forest, but does the problem refer to the final stage or an intermediate one?
On the other hand, B is primary succession, starting from a lava field, going through ㉡, and then to grassland. In the early stage of primary succession, pioneer species such as lichen or moss appear, and then herbaceous plants grow, followed by shrub forest, and eventually deciduous, mixed, and shade-tolerant forests in order. However, since B goes from lava field through ㉡ to grassland, it's likely that ㉡ is lichen. That's because in harsh environments like lava fields, what settles first tends to be things like lichen. Then ㉡ is lichen, and ㉠ would be deciduous forest. That's because, as A undergoes secondary succession, after going through shrub forest it would further develop into deciduous forest. Therefore, ㉠ is deciduous forest and ㉡ is lichen. Accordingly, statement ㄴ, "㉠ is deciduous forest," is a correct statement.
Next, let's look at figure (b). It shows changes in total productivity and respiration of plant community K depending on time (t). At time t0, it appears, and up to t1, total productivity increases, then afterwards decreases, while respiration continues to increase, and from t2 onward, it is marked as the appearance of a shade-tolerant forest. That is, before t1, total productivity is greater than respiration, so net productivity has a positive value, after t1, total productivity decreases and becomes similar to or even less than respiration, so net productivity reduces.
Net productivity is total productivity minus respiration, so at t1, total productivity is at its maximum and respiration is relatively low, so net productivity will be very high. But at t2, total productivity is lower and respiration continues to increase, so net productivity will be far lower than at t1. Therefore, the ratio of K's net productivity to respiration is greater at

t1 than at t2. But the choice ㄷ says, “K’s net productivity/respiration is greater at t2 than at t1,” so this is a wrong explanation. So ㄷ is wrong too.
In summary:
- ㄱ: “B is secondary succession.” → B is primary succession, so wrong.
- ㄴ: “㉠ is deciduous forest.” → Correct.
- ㄷ: net productivity/respiration is t2 > t1 → Wrong. t1 is greater.
Therefore, the answer is only ㄴ is correct, so we need to check if there is an option with only ㄴ. The options provided are as follows:
① ㄱ
② ㄴ
③ ㄱ, ㄷ
④ ㄴ, ㄷ
⑤ ㄱ, ㄴ, ㄷ
In this, the case where only ㄴ is correct is option ②, but let’s check again whether there was any calculation mistake.
At t1, total productivity is high and respiration is still low, so net productivity is abundant, and at t2, total productivity has decreased and respiration has increased, so net productivity is lower. Therefore, the ratio of net productivity to respiration, that is, net productivity/respiration, is much greater at t1. For example, if at t1, total productivity is 10 and respiration is 3, net productivity is 7 and the ratio is 7/3 2.33. If at t2, total productivity is 5 and respiration is 4, net productivity is 1 and the ratio is 1/4 = 0.25. So t1 is definitely greater. Thus, ㄷ is wrong, and the answer is option ②.
Final Answer: $\boxed{(2)}$ <|im_end|>
<|im_start|>assistant
Analyze figures (a) and (b) to solve the problem.
1. **Analysis of succession process**:
- **A**: Starts from grassland, passes through shrub forest, and proceeds to ㉠. Since it starts from grassland where plants already exist, it is **secondary succession**.
- **B**: Starts from lava field, passes through ㉡, and proceeds to grassland. Since it starts from a lava field with no soil, it is **primary succession**.
- Therefore, ㄱ ("B is secondary succession.") is an **incorrect** explanation.
2. **Identification of ㉠ and ㉡**:
- **A** (secondary succession): Passes through shrub forest and proceeds to ㉠, and generally, the late stage of secondary succession is a **deciduous forest**.
- **B** (primary succession): Starts from lava field, passes through ㉡, and proceeds to grassland, with the initial stage being **lichen**.
- Therefore, ㄴ ("㉠ is a deciduous forest.") is a **correct** explanation.
3. **Graph analysis**:
- **Total productivity** increases until t_1 and then decreases, while **respiration** continues to increase.
- **Net productivity** is at its highest at t_1 when total productivity is maximal and respiration is relatively low, so the ratio is high. At t_2, total productivity decreases and respiration increases, so the net productivity ratio becomes lower.
- Therefore, ㄷ ("K’s $\frac{\text{net productivity}}{\text{respiration}}$ is greater at t_2 than at t_1") is an **incorrect** explanation.
In conclusion, the only correct explanation is **ㄴ**.
$\boxed{(2)}$ <|im_end|>

HyperCLOVA X THINK (Original Model Output – Korean)

<|im_start|>assistant/think
좋아, 이 문제를 풀어보자. 일단 문제에서 주어진 그림을 잘 봐야겠지.
먼저 8번 문제의 그림 (가)와 (나)를 살펴봐야 해. 문제에 따르면 A와 B는 1차 천이와 2차 천이 중 하나이고, ㉠과 ㉡은 양수림과 지의류를 순서 없이 나타낸 거야. 그리고 식물군집 K의 시간에 따른 총생산량과 호흡량을 그래프로 보여주고 있어.

일단 (가)부터 보자. A는 초원에서 관목림을 거쳐 ㉠으로 가고, B는 용암 대지에서 시작해서 ㉡을 거쳐 초원으로 간다고 되어 있네. 여기서 A와 B가 각각 어떤 천이 과정인지 알아내야 해. 또 ㉠과 ㉡이 무엇인지도 파악해야 하고.
천이 과정을 생각해보면, 1차 천이는 생명이 없는 지역에서 시작하는 거고, 2차 천이는 기존에 식물이 있던 곳에서 교란이 일어난 후에 다시 회복되는 과정이야. 예를 들어 산불이나 벌목 같은 것들로 인해 생긴 빈 땅에서 시작하는 게 2차 천이지. 반면 1차 천이는 바위나 용암 대지처럼 토양조차 없는 곳에서 시작돼.
문제에서 A는 초원(이미 식물이 있는 곳)에서 시작하니까 이건 2차 천이 과정이겠지. 왜냐하면 초원은 이미 토양이 있고 일부 생물이 존재하는 환경이니까. 반면 B는 용암 대지에서 시작하는데, 용암 대지는 화산 활동으로 인해 형성된 곳으로 토양이 없을 가능성이 높아. 그래서 B는 1차 천이 과정에 해당할 거야. 따라서 A는 2차 천이, B는 1차 천이라는 결론이 나오지만... 잠깐, 문제에서는 A와 B가 1차와 2차를 순서없이 나타냈다고 했으니까 내가 반대로 생각했을 수도 있으니 확인 필요.
다시 보면, A: 초원 → 관목림 → ㉠
B: 용암대지 → ㉡ → 초원
A는 이미 초원이 있으므로 2차 천이 과정이고, B는 용암대지에서 시작하므로 1차 천이 과정이다. 그러면 A는 2차, B는 1차가 맞아. 그럼 보기에서 ㄱ은 'B는 2차 천이이다.'라고 했으니 틀린 설명이 되겠지. 그러니까 ㄱ은 틀렸어.
다음으로 ㉠과 ㉡이 뭔지 보자. A 과정은 2차 천이이므로, 일반적으로 2차 천이의 종착 단계는 극상인 음수림이 될 수 있지만, 경우에 따라서는 다른 형태일 수도 있겠지. 하지만 보통 2차 천이는 시간이 지나면 더 복잡한 구조의 숲으로 발달해. 그런데 여기서는 A가 관목림을 거쳐 ㉠으로 가는데, 만약 ㉠이 양수림이라면 그 다음 단계는 음수림일 텐데, 문제에서는 최종 단계를 나타내는 걸까? 아니면 중간 단계일까?
반면 B는 1차 천이로, 용암대지에서 시작하여 ㉡을 거쳐 초원으로 간다. 1차 천이의 초기에는 지의류나 선태식물 같은 개척자들이 나타나고, 그 다음에 초본 식물들이 자라면서 관목림이 되고, 결국에는 양수림, 혼합림, 음수림 순으로 진행되지. 그런데 B는 용암대지에서 ㉡을 거쳐 초원으로 가므로, ㉡은 아마도 지의류일 가능성이 높아. 왜냐하면 용암대지와 같은 척박한 환경에서 가장 먼저 정착하는 것은 지의류 같은 것이기 때문이야. 그렇다면 ㉡은 지의류이고, ㉠은 양수림이 되는 거겠지. 왜냐하면 A는 2차 천이 과정에서 관목림을 거쳐 더 발전하면 양수림이 될 테니까. 따라서 ㉠은 양수림, ㉡은 지의류가 돼. 그러므로 보기의 ㄴ은 '㉠은 양수림이다'라고 했으니 맞는 설명이야.
그 다음으로 (나) 그래프를 보자. 식물 군집 K의 시간(t)에 따른 총생산량과 호흡량의 변화인데, t0 시점에서 출현하고, t1까지는 총생산량이 증가하다가 이후 감소하며, 호흡량은 계속 증가하고, t2 시점부터는 음수림 출현이라고 표시되어 있어. 즉, t1 이전에는 총생산량이 호흡량보다 많아서 순생산량이 양의 값을 가지다가, t1 이후에는 총생산량이 감소하면서 호흡량과 비슷해지거나 오히려 작아져서 순생산량이 줄어들게 되지.
순생산량은 총생산량 - 호흡량이니까, t1 때는 총생산량이 최대이고 호흡량은 상대적으로 낮아서 순생산량이 매우 높을 거야. 그러나 t2 때가 되면 총생산량은 더 낮아지고 호흡량은 계속 증가해서 순생산량이 t1 때보다 훨씬 낮아질 거야. 따라서 K의 순생산물 / 호흡량 비율은 t1 때가 t2 때보다 크다는 결론이 나와. 그런데 문제의 보기 ㄷ은 'K의 순생산량/호흡량은 t2일 때가 t1일 때보다 크다'라고 했으므로 이는 틀린 설명이야. 따라서 ㄷ도 틀렸다.
정리하면:
- ㄱ: B는 2차 천이이다. → B는 1차 천이이므로 틀림.
- ㄴ: ㉠은 양수림이다. → 맞음.
- ㄷ: 순생산량/호흡량은 t2 > t1 → 틀림. t1이 더 큼.
따라서 정답은 ㄴ만 맞는 경우인데, 보기 중에 ㄴ만 있는 선택지가 있는지 확인해봐야 해.
문제에서 제시된 보기 옵션은 다음과 같아:
① ㄱ
② ㄴ
③ ㄱ, ㄷ
④ ㄴ, ㄷ
⑤ ㄱ, ㄴ, ㄷ
여기서 ㄴ만 맞는 경우는 ②번이지만, 혹시 계산 실수가 있었는지 다시 확인해보자.
t1 시기에는 총생산량이 높고 호흡량은 아직 낮은 상태라 순생산량이 많을 거고, t2 시기에는 총생산량이 감소했고 호흡량은 증가했으므로 순생산량은 더 적어져. 따라서 순생산량 대비 호흡량의 비율, 즉 순생산량/호흡량은 t1 때가 훨씬 높다. 예를 들어 t1에서 총생산량

이 10, 호흡량 3이면 순생산량은 7이고 비율은 7/3 2.33. t2에서 총생산량 5, 호흡량 4라면 순생산량 1, 비율은 1/4 = 0.25. 따라서 확실히 t1이 더 크다. 그러니 ㄷ은 틀렸고, 따라서 정답은 ②번.
Final Answer: $\boxed{(2)}$ <|im_end|>
<|im_start|>assistant
그림 (가)와 (나)를 분석하여 문제를 해결한다.
1. **천이 과정 분석**:
- **A**: 초원에서 시작하여 관목림을 거쳐 ㉠으로 진행된다. 이미 식물이 있는 초원에서 시작되므로 **2차 천이**이다.
- **B**: 용암 대지에서 시작하여 ㉡을 거쳐 초원으로 진행된다. 토양이 없는 용암 대지에서 시작하므로 **1차 천이**이다.
- 따라서 ㄱ("B는 2차 천이이다.")은 **틀린** 설명이다.
2. **㉠과 ㉡ 식별**:
- **A**(2차 천이): 관목림을 거쳐 ㉠으로 진행되며, 일반적으로 2차 천이의 후기 단계는 **양수림**이다.
- **B**(1차 천이): 용암 대지에서 시작하여 ㉡을 거쳐 초원으로 진행되며, 초기 단계는 **지의류**이다.
- 따라서 ㄴ("㉠은 양수림이다.")은 **맞는** 설명이다.
3. **그래프 분석**:
- **총생산량**은 t_1까지 증가한 후 감소하며, **호흡량**은 지속적으로 증가한다.
- **순생산량**은 t_1 시기에 총생산량이 최대이고 호흡량이 상대적으로 낮아 비율이 높다. t_2 시기에는 총생산량이 감소하고 호흡량이 증가하여 순생산량 비율이 낮아진다.
- 따라서 ㄷ("K의 $\frac{\text{순생산량}}{\text{호흡량}}$은 t_2일 때가 t_1일 때보다 크다")은 **틀린** 설명이다.
최종적으로 옳은 설명은 **ㄴ**뿐이다.
$\boxed{(2)}$ <|im_end|>

D.2 KCSAT Biology Advanced Example

We present our model's output on a sample KCSAT Biology Advanced problem. Figure 8 shows the original problem image, which is directly provided to the model for analysis and reasoning. The model is required to interpret, reason through, and solve the problem solely based on the image. For reference, we include the model's original answer, as well as an English translation of the problem for readers. A literal English translation of the model's output is also provided for comparison.

2. 표는 식물을 구성하는 조직계의 특징을 나타낸 것이다. A와 B는 관다발 조직계와 표피 조직계를 순서 없이 나타낸 것이다.

조직계	특징
A	표면을 덮어 식물체를 보호한다.
B	?
기본 조직계	양분을 합성하고 저장한다.

이에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보 기>

ㄱ. A는 표피 조직계이다.
ㄴ. B를 통해 물질이 이동한다.
ㄷ. 장미의 잎에는 기본 조직계가 있다.

① ㄱ　② ㄴ　③ ㄱ, ㄷ　④ ㄴ, ㄷ　⑤ ㄱ, ㄴ, ㄷ

Figure 8: Original KCSAT Sample (in Korean).

Sample Question 2 (Translated)
The table below summarizes the characteristics of plant tissues. A and B refer to vascular and dermal tissues, listed in no particular order.

Tissue	Characteristic
A	Covers the surface and protects the plant body.
B	?
Fundamental tissue	Synthesizes and stores nutrients.

Which statements below, from the options labeled (a) to (c), are correct regarding the tissues described above?

(Options)

(a) A is a dermal tissue.
(b) Substances are transported via tissue B.
(c) The petal of a rose contains fundamental tissue.

Select the correct combination(s):

(1) a (2) b (3) a, c (4) b, c (5) a, b, c

HyperCLOVA X THINK (Translated Model Output – English)

<|im_start|>assistant/think
Okay, let's solve this problem. First, I need to carefully read the question. The question gives a table showing the characteristics of plant tissue systems, and A and B are listed as vascular tissue system and epidermal tissue system in random order. Also, it is stated that the ground tissue system synthesizes and stores nutrients.
First, looking at the table, the characteristic of A is "covers the surface and protects the plant body," and B is not yet explained. However, since the ground tissue system is already described as synthesizing and storing nutrients, the remaining two must be the vascular and epidermal tissue systems. The problem states that one of A or B is epidermal, and the other is vascular.
Let's summarize the characteristics of each tissue system.
1. Ground tissue system: This is also known as the fundamental tissue and mainly consists of parenchyma and sclerenchyma, which do not perform photosynthesis but serve to store or transport water and nutrients. So, in the problem, the characteristic of the ground tissue system is given as "synthesizes and stores nutrients"... Wait, does the ground tissue system photosynthesize? No, usually it's tissues like palisade cells in the epidermal tissue or vascular tissue that do that, isn't it? Wait, this could be a confusing point. I was taught that the ground tissue system generally does not photosynthesize, but in the problem, its characteristic is "synthesizes and stores nutrients." I need to check if this is correct.
Is the problem wrong? Or am I misremembering?
The ground tissue system mainly stores, and photosynthesis occurs in the epidermal tissue (especially palisade cells below the leaf epidermis). So is the description of the ground tissue system in the problem incorrect? Or is there another interpretation?
Since the problem states that the ground tissue system "synthesizes and stores nutrients," I have no choice but to accept it as it is for now. Probably, that's how the problem defines it.
Going back to the question: is A the epidermal tissue system or vascular? The characteristic of A is "covers the surface and protects the plant body," which is a major function of the epidermal tissue system. For example, epidermal cells cover the surface of leaves, preventing water evaporation and protecting from pathogens. Therefore, A is epidermal tissue system, and B is likely vascular tissue system.
Then, what would be the feature of B? The vascular tissue system is composed of xylem and phloem and is involved in the transport of water and nutrients, so the characteristic of B would be something like "transports water and nutrients." But the characteristic of B is left blank in the problem, so we have to infer it and select the correct explanation from the choices.

Now let's look at the choices.
ㄱ. A is the epidermal tissue system. → Correct. Based on the reasoning above, this is true.
ㄴ. Substances move through B. → If B is the vascular tissue system, then it is the pathway for water and nutrient transport, so this is also true.
ㄷ. Rose leaves have ground tissue system. → The ground tissue system should exist in all plant leaves. As the internal tissue of leaves, it stores or transports water and nutrients, so this is also true.
However, the problem describes the ground tissue system as "synthesizes and stores nutrients," which makes me doubtful as to whether it really does photosynthesis, but since that is what the problem states, I have to accept it. So ㄷ is also a correct explanation.
But based on prior knowledge, the ground tissue system does not photosynthesize, so I wonder if the problem is in error. If, for example, the problem actually refers to the epidermal tissue system instead? No, the problem specifically describes the characteristic of the ground tissue system, so I should just follow that.
So are ㄱ, ㄴ, ㄷ all correct? The answer options include 5 as "ㄱ, ㄴ, ㄷ," so that seems possible, but is there any chance I made a mistake?
Let me check again.
A: Epidermal tissue system → covers surface. O
B: Vascular tissue system → transports water and nutrients. O
Ground tissue system: In problem, synthesizes and stores nutrients. Might be different from reality but will follow the problem.
Do rose leaves have ground tissue system? The inside of leaves is made of parenchyma, which is ground tissue, so O.
Therefore, it seems all three are correct. Is the answer 5?
But sometimes in tests there are traps—could ㄷ somehow be wrong?
For example, if rose leaves had only vascular tissues and not ground tissue? But in a leaf, there are epidermis, guard cells, palisade tissue (ground tissue), vascular bundles, etc., so ground tissue must be present. Thus, ㄷ is valid.
So the answer is 5?
Even though at first I thought the description of the ground tissue system was wrong, since that's how the problem gives it, I will follow it. Therefore, all ㄱ, ㄴ, and ㄷ are correct, and the answer is 5.
Final answer: $\boxed{(5)}$ <|im_end|>
<|im_start|>assistant
In the table for question 2, A and B represent the epidermal tissue system and vascular tissue system, respectively. The characteristic of the ground tissue system is given as 'synthesizes and stores nutrients.'
- **Characteristic of A**: 'Covers the surface and protects the plant body.' This is a main function of the epidermal tissue system, which covers the surface of the plant to prevent water evaporation and protect from external environments. Therefore, A is the epidermal tissue system.
- **Characteristic of B**: It is a characteristic of the vascular tissue system. The vascular tissue system is composed of xylem and phloem and is involved in the transport of water and nutrients. Therefore, B is the vascular tissue system, and it is correct to say that substances are transported through it.
- **Ground tissue system**: The characteristic given in the problem is 'synthesizes and stores nutrients.' The ground tissue system comprises most of the plant and serves to store and transport water and nutrients. The leaves of a rose also contain ground tissue system, carrying out these functions.
Analysis of choices:
- **ㄱ**: A is the epidermal tissue system. (Correct)
- **ㄴ**: Substances are transported through B. (Correct)
- **ㄷ**: The leaves of a rose have ground tissue system. (Correct)
Therefore, since all choices are correct, the answer is number 5.
$\boxed{(5)}$ <|im_end|>

HyperCLOVA X THINK (Original Model Output – Korean)

<|im_start|>assistant/think
좋아, 이 문제를 풀어보자. 우선 문제를 잘 읽어봐야겠어. 문제는 식물을 구성하는 조직계의 특징을 나타낸 표인데, A와 B는 관다발 조직계와 표피 조직계를 순서 없이 나열했다고 해. 그리고 기본 조직계는 양분을 합성하고 저장한다고 되어 있네.
먼저, 표를 보면 A의 특징은 '표면을 덮어 식물체를 보호한다'고 했고, B는 아직 설명이 안 되어 있어. 그런데 기본 조직계는 이미 양분 합성과 저장이라고 했으니까, 나머지 두 개가 관다발과 표피일 거야. 문제에서 A와 B 중 하나는 표피이고 다른 하나는 관다발이야.
그럼 먼저 각 조직계의 특징을 정리해보자.
1. **기본 조직계**: 이건 울타리라고도 불리는 조직으로, 주로 유조직과 기계 조직으로 이루어져 있고, 광합성을 하지 않지만 물과 영양분을 저장하거나 운반하는 역할을 한다고 알고 있어. 그래서 문제에서도 기본 조직계의 특징이 '양분을 합성하고 저장한다'고 했는데... 잠깐, 기본 조직계가 광합성을 하나? 아니, 보통 표피 조직이나 관다발 조직 중에 책상세포 같은 게 있지 않나?
잠깐, 여기서 헷갈리기 쉬운 부분이 있을 것 같아. 기본 조직계는 일반적으로 광합성을 하지 않는다고 배웠던 것 같은데, 문제에서는 기본 조직계의 특징이 '양분을 합성하고 저장한다'고 했어. 이게 맞는지 확인해야 할 것 같아.
아니면 문제가 잘못된 건가? 아니면 내가 잘못 기억한 건가?
기본 조직계는 주로 저장 기능을 하며, 광합성은 표피 조직(특히 잎의 표피 아래에 있는 책상세포)에서 일어나는 걸로 알고 있어. 그럼 문제의 기본 조직계 설명이 틀린 건가? 아니면 다른 해석이 있을까?
문제에서 기본 조직계의 특징이 '양분을 합성하고 저장한다'고 했으니 일단 그대로 받아들이고 넘어가야겠지. 아마 문제에서는 그렇게 정의했으니까.
그럼 다시 문제로 돌아가서, A는 표피 조직계일까 관다발일까? A의 특징은 '표면을 덮어 식물체를 보호한다'인데, 표피 조직계의 주요 기능 중 하나가 바로 식물체의 표면 덮개 역할이지. 예를 들어 잎의 표피 세포들이 표면을 덮어서 수분 증발을 막고 병원체로부터 보호해주니까. 따라서 A는 표피 조직계이고, B는 관다발 조직계일 가능성이 높아.
그러면 B의 특징은 뭘까? 관다발 조직계는 물관부와 체관부로 나뉘어져 있어서 물과 양분의 이동에 관여하니까, B의 특징은 아마도 '물과 양분을 운반한다' 정도가 될 것 같아. 하지만 문제에서는 B의 특징이 비어있고, 우리가 그걸 추론해서 보기 중에서 맞는 설명을 골라야 하니까.
이제 보기를 살펴보자.
ㄱ. A는 표피 조직계이다. → 맞아. 위에서 생각한 대로라면 맞는 말이야.
ㄴ. B를 통해 물질이 이동한다. → B가 관다발 조직계라면, 물과 양분이 이동하는 통로이므로 맞는 말이지.
ㄷ. 장미의 잎에는 기본 조직계가 있다. → 기본 조직계는 모든 식물의 잎에 존재할 거야. 잎의 내부 조직으로서 물과 양분을 저장하거나 운반하는 역할을 하니까. 따라서 이것도 맞는 말이야.
그런데 문제에서 기본 조직계의 특징이 '양분을 합성하고 저장한다'고 했는데, 실제로 기본 조직계가 광합성을 하는지 의문이 들지만, 문제에서는 그렇게 주어졌으므로 일단은 받아들여야 해. 그러면 ㄷ도 맞는 설명이 되겠지.
하지만 기존 지식으로는 기본 조직계가 광합성을 하지 않는다고 알고 있는데, 이 부분이 문제의 오류인지 아닌지가 궁금해. 만약 문제에서의 기본 조직계가 실제로는 표피 조직계를 가리키는 것이 아니라면? 아니, 문제에서는 기본 조직계의 특징을 따로 주었으니 그대로 따라가야 할 것 같아.
따라서 ㄱ, ㄴ, ㄷ 모두 옳은 설명인가? 그런데 보기에서 5번이 'ㄱ, ㄴ, ㄷ'이라서 그럴 수도 있지만, 혹시 내가 뭔가 잘못 생각했을까?
다시 한 번 점검해보자.
- A: 표피 조직계 → 표면 덮개 역할. O
- B: 관다발 조직계 → 물과 양분 운반. O
- 기본 조직계: 문제에서는 양분 합성 및 저장. 실제와 다를 수 있으나 문제에 따름.
장미의 잎에는 기본 조직계가 있는가? 잎의 내부는 유조직으로 구성되어 있는데, 이는 기본 조직계에 해당하므로 O.
그래서 세 가지 모두 맞는 것 같다. 그렇다면 정답은 5번?
하지만 보통 시험 문제에서는 이런 함정이 있을 수 있으니, 혹시 ㄷ가 틀린 경우는 없을까?

예를 들어, 장미의 잎에는 기본 조직계가 없고 관다발만 있다는 식으로? 아니, 잎의 경우 표피, 공변세포, 울타리 조직(기본 조직계), 관다발 등이 있으니까 기본 조직계가 있어야 해. 따라서 ㄷ는 맞는 설명.
그럼 답은 5번?
하지만 처음에 기본 조직계의 특징이 틀렸다는 생각이 들었지만, 문제에서는 그렇게 주어졌으니 따라가야 한다. 따라서 ㄱ, ㄴ, ㄷ 모두 맞는 설명이므로 5번이 정답이다.
최종 답변 $\boxed{(5)}$ <|im_end|>
<|im_start|>assistant
2번 문제의 표에서 A와 B는 각각 표피 조직계와 관다발 조직계를 나타낸다. 기본 조직계의 특징은 '양분을 합성하고 저장한다'로 주어져 있다.
- **A의 특징**: '표면을 덮어 식물체를 보호한다.' 이는 표피 조직계의 주요 기능으로, 표피 조직계는 식물체의 표면을 덮어 수분 증발을 막고 외부 환경으로부터 보호한다. 따라서 A는 표피 조직계이다.
- **B의 특징**: 관다발 조직계의 특징이다. 관다발 조직계는 물관과 체관으로 구성되어 물과 양분의 이동에 관여한다. 따라서 B는 관다발 조직계이며, 이를 통해 물질이 이동한다는 것은 올바른 설명이다.
- **기본 조직계**: 문제에서 주어진 특징은 '양분을 합성하고 저장한다.' 기본 조직계는 식물의 대부분을 차지하며, 물과 양분의 저장 및 운반 역할을 한다. 장미의 잎에도 기본 조직계가 존재하여 이러한 기능을 수행한다.
보기 분석: - **ㄱ**: A는 표피 조직계이다. (정답) - **ㄴ**: B를 통해 물질이 이동한다. (정답) - **ㄷ**: 장미의 잎에는 기본 조직계가 있다. (정답)
따라서 모든 선택지가 올바르므로 정답은 5번이다.
$\boxed{(5)}$ <|im_end|>

D.3 KCSAT Math Advanced Example

We present our model's output on a sample KCSAT Math Advanced problem. Figure 9 shows the original problem image, which is directly provided to the model for analysis and reasoning. The model is required to interpret, reason through, and solve the problem solely based on the image. For reference, we include the model's original answer, as well as an English translation of the problem for readers. A literal English translation of the model's output is also provided for comparison.

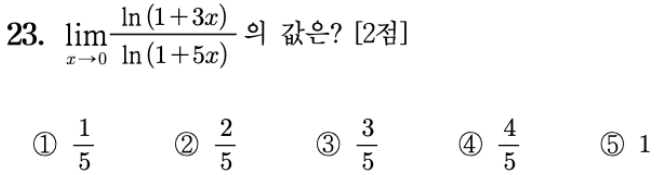


Figure 9: Original KCSAT Sample (in Korean).

Sample Question 23 (Translated)
Determine the value of

$$\lim_{x\to 0}\frac{\ln(1+3x)}{\ln(1+5x)}$$

Choose the correct answer: (1) $\frac{1}{5}$ (2) $\frac{2}{5}$ (3) $\frac{3}{5}$ (4) $\frac{4}{5}$ (5) 1

HyperCLOVA X THINK (Translated Model Output – English)

<|im_start|>assistant/think
Alright, this seems to be a problem about finding a limit. It's question 23, and it presents the ratio of logarithmic functions in fractional form. The expression is the limit as $x \to 0$ of $\frac{\ln(1+3x)}{\ln(1+5x)}$. The provided choices range from $\frac{1}{5}$ to 1, but which one is correct?

First, for limits of this type, you can usually use L'Hôpital's Rule to solve them, since as $x \to 0$, both the numerator and denominator become $\ln(1)$, so the expression is in the indeterminate form $0/0$. By applying L'Hôpital's Rule, we can find the answer by differentiating the numerator and denominator.
Before applying L'Hôpital's Rule, let's check: when $x = 0$, the numerator is $\ln(1 + 3 \cdot 0) = \ln(1) = 0$, and the denominator is $\ln(1 + 5 \cdot 0) = \ln(1) = 0$, so that checks out. Therefore, L'Hôpital's Rule can be used. So, we need to differentiate both the numerator and denominator with respect to x. The derivative of the numerator $\ln(1 + 3x)$ is $\frac{3}{1+3x}$, and the derivative of the denominator $\ln(1 + 5x)$ is $\frac{5}{1+5x}$.
By L'Hôpital's Rule, the limit then becomes

$$\frac{\frac{3}{1+3x}}{\frac{5}{1+5x}} = \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

and, simplifying this,

$$\frac{3(1+5x)}{5(1+3x)}$$

is what you get.
Now, if you calculate the value as $x \to 0$, $1 + 5x \to 1$, and $1 + 3x \to 1$, so overall it should become $\frac{3}{5}$, right? Looking at the choices, $\frac{3}{5}$ is choice ③. But why did I think of something else at first? Oh, did I calculate something incorrectly?
Let me double-check. After applying L'Hôpital's Rule:

$$\lim_{x \to 0} \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

Plugging in $x \to 0$,

$$\frac{3}{1} \cdot \frac{1}{5} = \frac{3}{5}$$

So, the answer is $\frac{3}{5}$? Among the answer choices, ③ is $\frac{3}{5}$, ② is $\frac{2}{5}$, and ④ is $\frac{4}{5}$, so I need to make sure not to get confused.
Can this be solved another way, for example, by approximating using a Taylor expansion?

$$\ln(1+a) \approx a - \frac{a^2}{2} + \cdots$$

Using this expansion, if a is small, we can approximate $\ln(1 + a) \approx a$.
So, in the numerator, $\ln(1 + 3x) \approx 3x$, and in the denominator, $\ln(1 + 5x) \approx 5x$, which makes the ratio

$$\frac{3x}{5x} = \frac{3}{5}$$

and we reach the same result.
So, since both methods yield $\frac{3}{5}$, it seems choice ③ is the answer. Is there any chance I made a calculation mistake in the process? Let's check again. When applying L'Hôpital's Rule, did I correctly differentiate the numerator and denominator:

$$\frac{d}{dx}\ln(1+3x) = \frac{3}{1+3x}, \quad \frac{d}{dx}\ln(1+5x) = \frac{5}{1+5x}$$

And when converting the division to multiplication, there was no sign or coefficient error, and as $x \to 0$, the x terms in the denominator and numerator disappear, so the result is indeed $\frac{3}{5}$. In the case of the Taylor expansion as well, considering only the first term, it's exactly $\frac{3x}{5x} = \frac{3}{5}$. So, certainly, $\frac{3}{5}$ is correct. Therefore, among the choices, since ③ is $\frac{3}{5}$, the final answer is ③.
Final answer: $\boxed{(3)}$ <|im_end|>
<|im_start|>assistant
To find the limit $\lim_{x \to 0} \frac{\ln(1+3x)}{\ln(1+5x)}$, we use the **L'Hôpital's Rule**.

Since both the numerator and denominator become the undetermined form $0/0$ as $x \to 0$, we can apply L'Hôpital's Rule:

$$\lim_{x\to 0} \frac{\ln(1+3x)}{\ln(1+5x)} = \lim_{x\to 0} \frac{\frac{d}{dx}\ln(1+3x)}{\frac{d}{dx}\ln(1+5x)}$$

Differentiating each:

$$\frac{d}{dx}\ln(1+3x) = \frac{3}{1+3x}, \quad \frac{d}{dx}\ln(1+5x) = \frac{5}{1+5x}$$

Therefore,

$$\lim_{x\to 0} \frac{\frac{3}{1+3x}}{\frac{5}{1+5x}} = \lim_{x\to 0} \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

When $x \to 0$, $1+3x \approx 1$ and $1+5x \approx 1$, so

$$\frac{3}{1} \cdot \frac{1}{5} = \frac{3}{5}$$

Also, using a Taylor expansion, $\ln(1+ax) \approx ax$ (where a is a constant) as an approximation:

$$\frac{\ln(1+3x)}{\ln(1+5x)} \approx \frac{3x}{5x} = \frac{3}{5}$$

Therefore, the answer is $\boxed{(3)}$.
<|im_end|>

HyperCLOVA X THINK (Original Model Output – Korean)

<|im_start|>assistant/think
좋아, 이 문제는 극한을 구하는 문제인 것 같아. 23번 문제고, 분수꼴의 로그 함수의 비율이 주어졌어. 식은 $x \to 0$일 때 $\frac{\ln(1+3x)}{\ln(1+5x)}$의 극한이야. 보기로는 $\frac{1}{5}$부터 1까지 있는데, 정답이 뭘까?
먼저, 이런 형태의 극한 문제는 보통 로피탈 법칙을 써서 풀 수 있지 않을까? 왜냐하면 분자와 분모 모두 $x \to 0$일 때 $\ln(1)$이 되니까 $0/0$ 형태의 불정형이기 때문이야. 로피탈 법칙을 적용하면 미분해서 계산할 수 있을 거야.
로피탈 법칙을 적용하기 전에 먼저 확인해보자. $x = 0$일 때 분자는 $\ln(1+3\cdot 0) = \ln(1) = 0$, 분모도 $\ln(1+5\cdot 0) = \ln(1) = 0$이니까 맞아. 따라서 로피탈 법칙을 사용할 수 있어. 그래서 분자와 분모를 각각 x로 미분해야 해. 분자인 $\ln(1+3x)$의 도함수는 $\frac{3}{1+3x}$이고, 분모인 $\ln(1+5x)$의 도함수는 $\frac{5}{1+5x}$이지.
그러면 로피탈 법칙에 의해 극한은

$$\frac{\frac{3}{1+3x}}{\frac{5}{1+5x}} = \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

가 되고, 이걸 단순화하면

$$\frac{3(1+5x)}{5(1+3x)}$$

가 되겠지.
이제 $x \to 0$일 때 이 값을 계산하면, $1+5x \to 1$, $1+3x \to 1$이니까 전체적으로 $\frac{3}{5}$가 되어야 하지 않나? 근데 보기에는 $\frac{3}{5}$가 ③번에 있네. 그런데 왜 내가 처음 생각했을 때 다른 결과가 나오는 걸까? 아, 아니, 계산을 잘못했나?
다시 확인해보자. 로피탈 법칙 적용 후의 식:

$$\lim_{x\to 0} \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

여기서 $x \to 0$을 대입하면,

$$\frac{3}{1} \cdot \frac{1}{5} = \frac{3}{5}$$

맞아, 그럼 답은 $\frac{3}{5}$인가? 그런데 문제에 주어진 보기 중에서 ③번이 $\frac{3}{5}$이고, ②번이 $\frac{2}{5}$, ④번이 $\frac{4}{5}$라서 헷갈리지 않게 해야겠어.
혹시 다른 방법으로 풀어보면, 예를 들어 테일러 전개를 사용해서 근사할 수 있을까?

$$\ln(1+a) \approx a - \frac{a^2}{2} + \cdots$$

이런 식으로 전개하면, a가 작을 때 $\ln(1+a) \approx a$ 정도로 근사할 수 있지.
그럼 분자는 $\ln(1+3x) \approx 3x$, 분모는 $\ln(1+5x) \approx 5x$가 되니까, 그 비율은

$$\frac{3x}{5x} = \frac{3}{5}$$

가 되어 동일한 결과를 얻을 수 있겠네.
그래서 두 방법 모두 $\frac{3}{5}$가 나오니까 정답은 ③번인 것 같다. 하지만 혹시나 계산 과정에서 실수한 부분이 있을까? 다시 한번 체크해보자. 로피탈 법칙 적용 시, 분자와 분모의 도함수를 제대로 했는지:

$$\frac{d}{dx}\ln(1+3x) = \frac{3}{1+3x}, \quad \frac{d}{dx}\ln(1+5x) = \frac{5}{1+5x}$$

그리고 나눗셈을 곱셈으로 바꿀 때 부호나 계수가 틀린 건 없고, $x \to 0$일 때의 값은 분모와 분자의 x 항이 없어지므로 $\frac{3}{5}$가 되는 게 맞지. 테일러 전개의 경우에도 첫 번째 항만 보면 정확히 $\frac{3x}{5x} = \frac{3}{5}$. 그래서 확실히 $\frac{3}{5}$가 맞는 것 같아. 그럼 문제의 보기 중에서 ③번이 $\frac{3}{5}$이므로 정답은 ③번이다.
최종 답변 $\boxed{(3)}$<|im_end|>
<|im_start|>assistant
극한 $\lim_{x\to 0} \frac{\ln(1+3x)}{\ln(1+5x)}$ 를 구하기 위해 **로피탈 법칙**을 사용합니다.
분자와 분모 모두 $x \to 0$ 일 때 $0/0$ 형태의 부정형이므로 로피탈 법칙을 적용합니다:

$$\lim_{x\to 0} \frac{\ln(1+3x)}{\ln(1+5x)} = \lim_{x\to 0} \frac{\frac{d}{dx}\ln(1+3x)}{\frac{d}{dx}\ln(1+5x)}$$

각각 미분하면:

$$\frac{d}{dx}\ln(1+3x) = \frac{3}{1+3x}, \quad \frac{d}{dx}\ln(1+5x) = \frac{5}{1+5x}$$

따라서,

$$\lim_{x\to 0} \frac{\frac{3}{1+3x}}{\frac{5}{1+5x}} = \lim_{x\to 0} \frac{3}{1+3x} \cdot \frac{1+5x}{5}$$

$x \to 0$ 일 때 $1+3x \approx 1$ 및 $1+5x \approx 1$ 이므로,

$$\frac{3}{1} \cdot \frac{1}{5} = \frac{3}{5}$$

또한 테일러 전개를 사용하면 $\ln(1+ax) \approx ax$ (단, a 는 상수)로 근사할 수 있습니다:

$$\frac{\ln(1+3x)}{\ln(1+5x)} \approx \frac{3x}{5x} = \frac{3}{5}$$

따라서, 답은 $\boxed{(3)}$ 입니다.
<|im_end|>

E Synthetic Data Examples

Seed Data	Synthetic QA Data
수학에서, 어떤 수의 제곱근(-根, 영어: square root) 또는 자승근은 제곱하여 그 수가 되는 모든 수이다. 거듭제곱의 역연산이다. 즉, 실수 및 복소수 a {\displaystyle a}에서, '제곱한 수 a {\displaystyle a}의 뿌리가 되는 모든 수'를 뜻한다. 실수의 범위에서만 보면, 모든 양의 실수는 서로 덧셈 역원인 두 제곱근을 가지며, 이 중 음이 아닌 하나를 주요 제곱근(主要 제곱根, 영어: principal square root)이라고 한다. 그러나 0의 제곱근은 0뿐이므로 이를 주요 제곱근으로 삼으며, 음의 실수의 실수 제곱근은 존재하지 않으므로 주요 제곱근을 정의할 수 없다. 예를 들어, 실수 9의 제곱근은 ±3이며, 이 중 주요 제곱근은 3이다. 또한 −4의 제곱근은 존재하지 않는다. 복소수의 범위에서 보면, 모든 0이 아닌 복소수는 서로 중심 대칭인 두 제곱근을 가지며, 이 중 편각이 원래의 반인 하나를 주요 제곱근으로 삼는다. 예를 들어, 복소수 2 i {\displaystyle 2i}의 제곱근은 ± (1 + i) {\displaystyle \pm (1+i)}이며, 이 중 주요 제곱근은 1 + i이다. 미지수 x {\displaystyle x}의 주요 제곱근은 근호를 써서 x {\displaystyle {\sqrt {x}}} (√x)라고 적고, x {\displaystyle x}의 제곱근 이라고 읽는다. source : wikipedia	**Q1:** 실수 9의 제곱근은 무엇인가요? 주요 제곱근은 무엇인가요? **A1:** 실수 9의 제곱근은 ±3이며, 이 중 음이 아닌 3이 주요 제곱근입니다. **Q2:** 복소수 2i의 제곱근은 무엇인가요? 주요 제곱근은 무엇인가요? **A2:** 복소수 2i의 제곱근은 ±(1+i)이며, 이 중 편각이 원래의 반인 1+i가 주요 제곱근입니다. **Q3:** 음의 실수 -4의 제곱근은 존재하나요? 존재한다면 어떻게 표현하나요? **A3:** 음의 실수 -4의 제곱근은 실수 범위에서는 존재하지 않습니다. 하지만 복소수 범위에서는 존재하며, ±2i로 표현됩니다.

Figure 10: Synthetic QA data generated from Wikipedia

Seed Data	Synthetic QA Data
In mathematics, the square root of a number is any number that, when squared, becomes that number. It is the inverse operation of exponentiation. That is, for real and complex numbers a, it means "all numbers whose square is a." Within the real number system, every positive real number has two square roots that are additive inverses of each other, and the non-negative one is called the principal square root. However, the square root of 0 is only 0, so it is considered the principal square root. Negative real numbers do not have real square roots, so a principal square root cannot be defined. For example, the square roots of the real number 9 are ±3, and the principal square root is 3. Also, −4 has no square root in the real number system. In the complex number system, every nonzero complex number has two square roots that are symmetric with respect to the origin, and the one whose argument is half of the original is called the principal square root. For example, the square roots of the complex number 2i are ±(1+i), and the principal square root is 1+i. The principal square root of an unknown x is written using the radical sign as √x and read as "the square root of x." source : wikipedia	**Q1:** What are the square roots of the real number 9? What is the principal square root? **A1:** The square roots of the real number 9 are ±3, and the non-negative 3 is the principal square root. **Q2:** What are the square roots of the complex number 2i? What is the principal square root? **A2:** The square roots of the complex number 2i are ±(1+i), and the one with half the original argument, 1+i, is the principal square root. **Q3:** Does the negative real number -4 have a square root? If so, how is it expressed? **A3:** The square root of the negative real number -4 does not exist in the real number system. However, in the complex number system, it exists and is expressed as ±2i.

Figure 11: Translated sample example of synthetic QA data generated from Wikipedia

Seed Data	Synthetic Data
1. 인공신경망 소개 - 인공신경망이란 - 인공신경망의 중요성 - 딥러닝의 등장 배경 2. 딥러닝 기본개념 - 뉴런과 레이어 - 활성화 함수 - 손실 함수 - 역전파 알고리즘 3. 컴퓨터 비전 - 이미지 처리 기초 - 컨볼루셔널 신경망 (CNN) - 이미지 분류 - 객체 검출 - 세그멘테이션 Source : Korea Education and Research Information Service	# 인공신경망과 딥러닝: 컴퓨터 비전의 세계 ## 1. 인공신경망 소개 ### 인공신경망이란 인공신경망(Artificial Neural Network, ANN)은 인간의 뇌 신경망을 모방한 계산 모델입니다. 생물학적 신경세포(뉴런)의 연결 구조를 단순화한 노드(뉴런)들이 계층적으로 연결되어 정보를 처리합니다. 이러한 인공 뉴런들은 입력값을 받아 가중치를 곱하고 활성화 함수를 적용하여 출력값을 생성합니다. 인공신경망의 핵심 요소는 다음과 같습니다: - **뉴런(Neuron)**: 정보 처리 단위 - **가중치(Weight)**: 입력 신호의 중요도를 조절하는 값 - **바이어스(Bias)**: 모델의 유연성을 높이는 상수 - **활성화 함수(Activation Function)**: 비선형성을 추가하는 함수 ### 인공신경망의 중요성 인공신경망은 현대 인공지능의 기반이 되는 핵심 기술입니다. 그 중요성은 다음과 같은 측면에서 나타납니다: 1. **패턴 인식 능력**: 복잡한 데이터 패턴을 학습하고 예측할 수 있습니다. 2. **비선형 문제 해결**: 선형 모델로는 해결할 수 없는 복잡한 문제를 다룰 수 있습니다. 3. **적응성과 유연성**: 다양한 유형의 데이터에 적용 가능합니다. 4. **병렬 처리 능력**: 대규모 병렬 처리를 통해 빠른 계산이 가능합니다. ### 딥러닝의 등장 배경 딥러닝(Deep Learning)은 인공신경망의 심화된 형태로, 다수의 은닉층(hidden layer)을 가진 네트워크를 의미합니다. 딥러닝의 등장은 다음과 같은 기술적 발전과 함께 이루어졌습니다: 1. **컴퓨팅 파워 향상**: GPU와 TPU 같은 고성능 하드웨어의 발전 2. **빅데이터의 등장**: 대규모 학습 데이터를 활용한 효과적인 학습 가능 3. **알고리즘 개선**: ReLU, Batch Normalization 등 효율적인 학습 기법 개발 4. **자동화 도구**: TensorFlow, PyTorch 등 딥러닝 프레임워크의 등장 ## 2. 딥러닝 기본개념 ### 뉴런과 레이어…(후략)

Figure 12: Synthetic educational data generated from course syllabus

Seed Data	Synthetic Data
1. Introduction to Artificial Neural Networks - What is an Artificial Neural Network - Importance of Artificial Neural Networks - Background of the Emergence of Deep Learning 2. Basic Concepts of Deep Learning - Neurons and Layers - Activation Functions - Loss Functions - Backpropagation Algorithm 3. Computer Vision - Basics of Image Processing - Convolutional Neural Networks (CNN) - Image Classification - Object Detection - Segmentation Source : Korea Education and Research Information Service	# Artificial Neural Networks and Deep Learning: The World of Computer Vision ## 1. Introduction to Artificial Neural Networks ### What is an Artificial Neural Network? An Artificial Neural Network (ANN) is a computational model inspired by the neural networks of the human brain. It consists of layers of nodes (neurons) that simplify the structure of biological neurons and are interconnected to process information. These artificial neurons receive inputs, apply weights, and generate outputs through an activation function. The key components of an artificial neural network are: * **Neuron**: The basic unit for information processing * **Weight**: A value that adjusts the importance of each input signal * **Bias**: A constant that enhances the model's flexibility * **Activation Function**: A function that introduces non-linearity ### Importance of Artificial Neural Networks Artificial neural networks are the foundation of modern artificial intelligence. Their significance can be seen in the following aspects: 1. **Pattern Recognition**: Capable of learning and predicting complex data patterns 2. **Non-linear Problem Solving**: Able to tackle problems that linear models cannot solve 3. **Adaptability and Flexibility**: Applicable to a wide variety of data types 4. **Parallel Processing Capability**: Enables fast computation through large-scale parallelism ### Background of the Emergence of Deep Learning Deep learning is an advanced form of artificial neural networks, characterized by networks with multiple hidden layers. The rise of deep learning has been driven by several technological advancements: 1. **Improved Computing Power**: The development of high-performance hardware such as GPUs and TPUs 2. **The Advent of Big Data**: The availability of massive datasets enabling effective training 3. **Algorithmic Improvements**: Development of efficient learning techniques like ReLU and Batch Normalization 4. **Automated Tools**: Emergence of deep learning frameworks such as TensorFlow and PyTorch ## 2. Basic Concepts of Deep Learning ### Neurons and Layers … (continued)

Figure 13: Translated sample example of synthetic educational data generated from course syllabus